

File created: 24-Apr-95 15:10:41 {DSK}<lispcore>library>new>TEDITMENU.;1

changes to: (FNS \TEDIT.TABTYPE.SET)

previous date: 25-Aug-94 10:54:56 {DSK}<king>export>lispcore>library>TEDITMENU.;4

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1983, 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991, 1992, 1993, 1994, 1995 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **TEDITMENUMCOMS**

[(FILES TEDITDCL)

(DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS (\SCRATCHLEN 64))
(FILES (LOADCOMP)
TEDITDCL))

[COMS ; Simple Menu Button support

(FNS MB.BUTTONEVENTINFN MB.DISPLAY MB.SETIMAGE MB.SELFN MB.SIZEFN MB.WHENOPERATEDFN MB.COPYFN
MB.GETFN MB.PUTFN MB.SHOWSELFN MBUTTON.CREATE MBUTTON.CHANGENAME MBUTTON.FIND.BUTTON
MBUTTON.FIND.NEXT.BUTTON MBUTTON.FIND.NEXT.FIELD MBUTTON.INIT MBUTTON.NEXT.FIELD.AS.NUMBER
MBUTTON.NEXT.FIELD.AS.PIECES MBUTTON.NEXT.FIELD.AS.TEXT MBUTTON.NEXT.FIELD.AS.ATOM
MBUTTON.SET.FIELD MBUTTON.SET.NEXT.FIELD MBUTTON.SET.NEXT.BUTTON.STATE TEDITMENU.STREAM
\TEDITMENU.SELSCREENER)

(GLOBALVARS MBUTTONIMAGEFNS)

(DECLARE%: DONTEVAL@LOAD DOCOPY (P (MBUTTON.INIT))

(ADDVARS (IMAGEOBJTYPES (TEditMenuButton FILE TEDITMENU GETFN MB.GETFN)

[COMS ;; Three-state (ON-OFF-NEUTRAL) menu buttons, for, e.g., character properties like BOLD

(FNS MB.CREATE.THREESTATEBUTTON MB.THREESTATE.DISPLAY MB.THREESTATE.SHOWSELFN
MB.THREESTATE.WHENOPERATEDFN MB.THREESTATEBUTTON.FN THREESTATE.INIT)

(DECLARE%: DONTEVAL@LOAD DOCOPY (P (THREESTATE.INIT))

[COMS ; One-of-N Menu button sets

(FNS MB.CREATE.NWAYBUTTON MB.NB.DISPLAYFN MB.NB.WHENOPERATEDFN MB.NB.SIZEFN MB.NWAYBUTTON.SELFN
MB.NWAYMENU.NEWBUTTON NWAYBUTTON.INIT MB.NB.PACKITEMS MB.NWAYBUTTON.ADDITEM)

(GLOBALVARS NWAYBUTTONIMAGEFNS)

(DECLARE%: DONTEVAL@LOAD DOCOPY (P (NWAYBUTTON.INIT))

(ADDVARS (IMAGEOBJTYPES (NWayButton FILE TEDITMENU GETFN MB.GETFN)

[COMS ;; Two-state, toggling menu buttons.

(FNS \TEXTMENU.TOGGLE.CREATE \TEXTMENU.TOGGLE.DISPLAY \TEXTMENU.TOGGLE.SHOWSELFN
\TEXTMENU.TOGGLE.WHENOPERATEDFN \TEXTMENU.TOGGLEFN \TEXTMENU.TOGGLE.INIT \TEXTMENU.SET.TOGGLE
)

(GLOBALVARS \TOGGLEIMAGEFNS)

(DECLARE%: DONTEVAL@LOAD DOCOPY (P (\TEXTMENU.TOGGLE.INIT))

(ADDVARS (IMAGEOBJTYPES (ToggleButon FILE TEDITMENU GETFN MB.GETFN)

[COMS ;; Margin Setting and display

(FNS DRAWMARGINSCALE MARGINBAR MARGINBAR.CREATE MB.MARGINBAR.SELFN MB.MARGINBAR.SIZEFN
MB.MARGINBAR.DISPLAYFN MDESCALE MSCALE MB.MARGINBAR.SHOWTAB MB.MARGINBAR.TABTRACK
\TEDIT.TABTYPE.SET MARGINBAR.INIT)

(BITMAPS \TEDIT.LEFTTAB \TEDIT.CENTERTAB \TEDIT.RIGHTTAB \TEDIT.DECIMALTAB \TEDIT.DOTTED.LEFTTAB
\TEDIT.DOTTED.CENTERTAB \TEDIT.DOTTED.RIGHTTAB \TEDIT.DOTTED.DECIMALTAB
TEDIT.EXTENDEDRIGHTMARK)

(GLOBALVARS MARGINBARIMAGEFNS)

(DECLARE%: DONTEVAL@LOAD DOCOPY (P (MARGINBAR.INIT))

(ADDVARS (IMAGEOBJTYPES (MarginRuler FILE TEDITMENU GETFN MB.GETFN)

[COMS ;; Text menu creation and support

(FNS \TEXTMENU.START \TEXTMENU.DOC.CREATE TEXTMENU.CLOSEFN)

(BITMAPS TEXTMENUICON TEXTMENUICONMASK)

[COMS ; TEdit-specific support

(FNS \TEDITMENU.CREATE \TEDIT.EXPANDED.MENU MB.DEFAULTBUTTON.FN \TEDITMENU.RECORD.UNFORMATTED
MB.DEFAULTBUTTON.ACTIONFN)

(FNS \TEDIT.CHARLOOKSMENU.CREATE \TEDIT.EXPANDEDCHARLOOKS.MENU \TEDIT.APPLY.BOLDNESS
\TEDIT.APPLY.CHARLOOKS \TEDIT.APPLY.OLINE \TEDIT.SHOW.CHARLOOKS \TEDIT.NEUTRALIZE.CHARLOOKS
\TEDIT.FILL.IN.CHARLOOKS.MENU \TEDIT.NEUTRALIZE.CHARLOOKS.MENU \TEDIT.PARSE.CHARLOOKS.MENU
\TEDIT.APPLY.SLOPE \TEDIT.APPLY.STRIKEOUT \TEDIT.APPLY.ULINE)

(FNS \TEDITPARAMENU.CREATE \TEDIT.EXPANDEDPARA.MENU \TEDIT.APPLY.PARALOOKS \TEDIT.SHOW.PARALOOKS
\TEDIT.NEUTRALIZE.PARALOOKS.MENU \TEDIT.RECORD.TABLEADERS)

(FNS \TEDIT.SHOW.PAGEFORMATTING \TEDITPAGEMENU.CREATE \TEDIT.APPLY.PAGEFORMATTING
TEDIT.UNPARSE.PAGEFORMAT)

[COMS ; Initialization Code

(GLOBALVARS TEDIT.EXPANDED.MENU TEDIT.EXPANDEDPARA.MENU TEDIT.CHARLOOKS.MENU
TEDIT.MENUDIVIDER.SPEC TEDIT.EXPANDEDMENU.SPEC TEDIT.CHARLOOKSMENU.SPEC
TEDIT.PARAMENU.SPEC TEDIT.PAGEMENU.SPEC TEDIT.EXPANDED.PAGEMENU)

(FNS \TEDIT.MENU.INIT)

(DECLARE%: DONTEVAL@LOAD DOCOPY (P (\TEDIT.MENU.INIT)

(\TEDITMENU.CREATE)

(\TEDIT.CHARLOOKSMENU.CREATE)

(\TEDITPARAMENU.CREATE)

(\TEDITPAGEMENU.CREATE)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)

(NLAML)
(LAMA]

```
(FILESLOAD TEDITDCL)
(DECLARE%: EVAL@COMPILE DONTCOPY
(DECLARE%: EVAL@COMPILE
(RPAQQ \SCRATCHLEN 64)
(CONSTANTS (\SCRATCHLEN 64)
)
(FILESLOAD (LOADCOMP)
TEDITDCL)
)
```

:: Simple Menu Button support

(DEFINEQ

(MB.BUTTONEVENTINFN

[LAMBDA (OBJ STREAM SEL RELX RELY SELWINDOW TEXTSTREAM) ; Edited 30-May-91 22:15 by jds

(* There was a buttn event inside a menu button. Make sure that the button gets turned OFF when the mouse moves outside it.)

```
(PROG [(OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX)
(replace (SELECTION SELKIND) of SEL with 'VOLATILE)
(COND
((IMAGEOBJPROP OBJ 'MENUBUTTON.SELECTED)
```

(* This button is still active from an earlier hit. Don't let it be selected again.)

```
(RETURN 'DON'T)
((AND (IGEQ RELX 0)
(IGEQ RELY 0)
(ILEQ RELX (fetch XSIZE of OBJBOX))
(ILEQ RELY (fetch YSIZE of OBJBOX)))
be left alone.)
(RETURN T))
(T
```

(* We're really inside the thing. Return an indication that we're

(* He's moved outside the button. Don't permit the selection.)

```
(RETURN 'DON'T])
```

(MB.DISPLAY

[LAMBDA (OBJ STREAM MODE) ; Edited 11-Jan-89 16:58 by jds

:: Display the innards of a menu button

```
(SELECTQ (IMAGESTREAMTYPE STREAM)
(DISPLAY ;: Going to the display. Use the cached bitmap version of the button
```

```
[PROG (BITMAP DS (OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX))
(X (DSPXPOSITION NIL STREAM))
(Y (DSPYPOSITION NIL STREAM)))
[SETQ BITMAP (COND
```

```
((IMAGEOBJPROP OBJ 'BITCACHE))
(T (MB.SETIMAGE OBJ)
(IMAGEOBJPROP OBJ 'BITCACHE)
[BITBLT BITMAP 0 0 STREAM X (SETQ Y (IDIFFERENCE Y (fetch YDESC of OBJBOX)
; Display the button's image
```

```
(COND
((EQ (IMAGEOBJPROP OBJ 'STATE)
'ON) ; If the button is ON, mark it so.
(BITBLT NIL 0 0 STREAM X Y (fetch XSIZE of OBJBOX)
(fetch YSIZE of OBJBOX)
'TEXTURE
'INVERT BLACKSHADE]))
```

```
(PROG (BITMAP DS (FONT (IMAGEOBJPROP OBJ 'MBFONT))
(TEXT (IMAGEOBJPROP OBJ 'MBTEXT))
OLOOKS) ; Going to some output image stream. Use the actual text.
(SETQ OLOOKS (DSPFONT (FONTCOPY FONT 'DEVICE STREAM)
STREAM)) ; Change to the font for this menu button.
(PRIN1 TEXT STREAM) ; Print the button text
(DSPFONT OLOOKS STREAM) ; And put the font back as it was.
```

)])

(MB.SETIMAGE

[LAMBDA (OBJ) (* jds "23-Aug-84 13:22")

```
(PROG ((MBFONT (IMAGEOBJPROP OBJ 'MBFONT))
(MBTEXT (IMAGEOBJPROP OBJ 'MBTEXT))
BOX BITMAP DS)
(SETQ BOX (create IMAGEBOX
```

```

        XSIZE _ (STRINGWIDTH MBTEXT MBFONT)
        YSIZE _ (FONTPROP MBFONT 'HEIGHT)
        YDESC _ (FONTPROP MBFONT 'DESCENT)
        XKERN _ 0))
    (SETQ BITMAP (BITMAPCREATE (fetch XSIZE of BOX)
                             (fetch YSIZE of BOX)))
    (IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
    (IMAGEOBJPROP OBJ 'BOUNDBOX BOX)
    (SETQ DS (DSPCREATE BITMAP))
    (DSPXOFFSET 0 DS)
    (DSPYOFFSET 0 DS)
    (DSPFONT MBFONT DS)
    (MOVETO 0 (FONTPROP MBFONT 'DESCENT)
            DS)
    (PRIN1 MBTEXT DS)
    (RETURN OBJ])

```

(MB.SELFN

```
[LAMBDA (OBJ SEL W FN)
```

; Edited 30-May-91 22:15 by jds
(* Calls a menu-button's associated function, then turns off the highlighting of the menu button.)

```

    (PROG [(TSEL (create SELECTION))
          (BUTTONFN (OR FN (IMAGEOBJPROP OBJ 'MBFN)
                       (\COPYSEL SEL TSEL)
                       (replace (SELECTION SELKIND) of SEL with 'CHAR)
                       (replace (SELECTION SET) of SEL with NIL)
                       (replace (SELECTION ONFLG) of SEL with NIL)
                       (COND
                        ((NEQ (AND BUTTONFN (APPLY* BUTTONFN OBJ SEL W)
                                   'DON'T)
                             (\FIXSEL TSEL (fetch (SELECTION \TEXTOBJ) of TSEL)
                                       (\SHOWSEL TSEL NIL NIL))))
                        )
          ])
```

(* Save the selection that points to the menu button.)
(* Call the button's function)
(* If the button fn left the selection alone,)
(* Turn off the button hilite)

(MB.SIZEFN

```
[LAMBDA (OBJ STREAM CURX RIGHTMARGIN)
```

(* jds "30-Aug-84 11:24")
(* Tell the size of a menu button)

```

    (PROG ((FONT (IMAGEOBJPROP OBJ 'MBFONT))
          BOX)
    [COND
      ((DISPLAYSTREAMP STREAM)
       )
      [(EQ 'INTERPRESS (IMAGESTREAMTYPE STREAM))
       (SETQ FONT (FONTCOPY FONT 'DEVICE 'INTERPRESS)
                (EQ 'PRESS (IMAGESTREAMTYPE STREAM))
                (SETQ FONT (FONTCOPY FONT 'DEVICE 'PRESS)
                (SETQ BOX (create IMAGEBOX
                                   XSIZE _ (STRINGWIDTH (IMAGEOBJPROP OBJ 'MBTEXT)
                                                         FONT)
                                   YSIZE _ (FONTPROP FONT 'HEIGHT)
                                   YDESC _ (FONTPROP FONT 'DESCENT)
                                   XKERN _ 0))
                (IMAGEOBJPROP OBJ 'BOUNDBOX BOX)
                (RETURN BOX])

```

(* We're formatting for the DISPLAY)

(MB.WHENOPERATEDFN

```
[LAMBDA (OBJ DS OPERATION SEL)
```

(* jds " 7-Feb-84 14:20")

```

    (SELECTQ OPERATION
      (HIGHLIGHTED (MB.SHOWSELFN OBJ SEL T DS))
      (UNHIGHLIGHTED
       (MB.SHOWSELFN OBJ SEL NIL DS))
      (SELECTED (MB.SELFN OBJ SEL DS))
      (DESELECTED)
      NIL])

```

(MB.COPYFN

```
[LAMBDA (OBJ)
```

(* jds "23-May-84 11:32")
(* Copy a menu button object.)

```

    (create IMAGEOBJ
      OBJECTDATUM _ (COPY (fetch (IMAGEOBJ OBJECTDATUM) of OBJ))
      IMAGEOBJPLIST _ (COPY (fetch (IMAGEOBJ IMAGEOBJPLIST) of OBJ))
      IMAGEOBJFNS _ (fetch (IMAGEOBJ IMAGEOBJFNS) of OBJ])

```

(MB.GETFN

```
[LAMBDA (OBJ FILE)
```

; Edited 20-Aug-87 16:17 by jds
(* READ a menu button from a file.)

```

    (ERROR)
    (PROG [(TEXT (IMAGEOBJPROP OBJ 'MBTEXT))
          (MBFN (IMAGEOBJPROP OBJ 'MBFN))
          (FONT (IMAGEOBJPROP OBJ 'MBFONT)
               (\STRINGOUT FILE TEXT)
               (\ATMOUT FILE MBFN)

```

```
(\ATMOUT FILE (FONTPROP FONT 'FAMILY))
(\SMALLPOUT FILE (FONTPROP FONT 'SIZE))
(for ATTR in (FONTPROP FONT 'FACE) do (\ATMOUT FILE ATTR])
```

(MB.PUTFN

[LAMBDA (OBJ FILE) ; Edited 20-Aug-87 16:17 by jds

;; Write a menu button from a file; suitable for re-reading using the image objects GETFN.

```
(PROG [(TEXT (IMAGEOBJPROP OBJ 'MBTEXT))
(MBFN (IMAGEOBJPROP OBJ 'MBFN))
(FONT (IMAGEOBJPROP OBJ 'MBFONT])
(HELP)
(\STRINGOUT FILE TEXT) ; The button's image
(\ATMOUT FILE MBFN) ; The FN called when hit
(\ATMOUT FILE (FONTPROP FONT 'FAMILY))
(\SMALLPOUT FILE (FONTPROP FONT 'SIZE))
(for ATTR in (FONTPROP FONT 'FACE) do (\ATMOUT FILE ATTR])
```

(MB.SHOWSELFN

[LAMBDA (OBJ SEL ON DS) ; Edited 11-Jan-89 16:35 by jds

(LET [(OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX)
(OR (IMAGEOBJPROP OBJ 'BITCACHE)
(MB.DISPLAY OBJ)) ; MAKE SURE THE DISPLAY FORM EXISTS

```
(BITBLT (IMAGEOBJPROP OBJ 'BITCACHE)
0 0 DS 0 0 (fetch (IMAGEBOX XSIZE) of OBJBOX)
(fetch (IMAGEBOX YSIZE) of OBJBOX)
'INPUT
'REPLACE)
(COND
((OR ON (EQ (IMAGEOBJPROP OBJ 'STATE)
'ON))
(BITBLT NIL 0 (fetch (IMAGEBOX YDESC) of OBJBOX)
DS 0 0 (fetch (IMAGEBOX XSIZE) of OBJBOX)
(fetch (IMAGEBOX YSIZE) of OBJBOX)
'TEXTURE
'INVERT BLACKSHADE]))
```

(MBUTTON.CREATE

[LAMBDA (MBTEXT MBFN MBFONT IMAGEFNS) ; Edited 11-Jan-89 16:10 by jds

;; Create a MENU BUTTON image object, and fill in its image and function-hook fields

```
(LET* ([REAL-FONT (OR MBFONT (FONTCLASSCOMPONENT DEFAULTFONT 'DISPLAY]
(OBJ (IMAGEOBJCREATE NIL (OR IMAGEFNS MBUTTONIMAGEFNS)))
(BOX (create IMAGEBOX
XSIZE _ (STRINGWIDTH MBTEXT REAL-FONT)
YSIZE _ (FONTPROP REAL-FONT 'HEIGHT)
YDESC _ (FONTPROP REAL-FONT 'DESCENT)
XKERN _ 0))
BITMAP DS)
(IMAGEOBJPROP OBJ 'MBFN MBFN) ; The function to be called when the button is pushed
(IMAGEOBJPROP OBJ 'MBTEXT MBTEXT) ; The text displayed in the button
(IMAGEOBJPROP OBJ 'MBFONT REAL-FONT) ; The font that text appears in
(MB.SETIMAGE OBJ) ; Set up the image for the button, so we don't create it
repeatedly.
OBJ])
```

(MBUTTON.CHANGENAME

[LAMBDA (TEXTOBJ OBJ NEWNAME) (* jds "23-Aug-84 13:26")

(* Change the text that appears in a button, and redisplay the button if it's visible)

```
(PROG (BOX BITMAP DS)
(IMAGEOBJPROP OBJ 'MBTEXT NEWNAME)
(MB.SETIMAGE OBJ)
(TEDIT.OBJECT.CHANGED TEXTOBJ OBJ))
```

(MBUTTON.FIND.BUTTON

[LAMBDA (LABEL TEXTSTREAM CH#) ; Edited 22-Apr-93 15:40 by jds
(* "27-Sep-84 00:52" gbn)

(* * returns the piece no containing the imageobj with MBTEXT prop LABEL)

```
(PROG ((LABELATOM (MKATOM LABEL))
OBJ STARTPCNO (PCTB (fetch (TEXTOBJ PCTB) of (TEXTOBJ TEXTSTREAM)))
START-OF-PIECE PC)
(RETURN (first (SETQ PC (\CHTOPC (OR CH# 1)
PCTB T))
while (AND PC (NOT (ATOM PC))) do (SETQ OBJ (fetch (PIECE POBJ) of PC)
(COND
([AND OBJ (EQ LABELATOM (MKATOM (IMAGEOBJPROP
OBJ
'MBTEXT])
```

```
(RETURN PCNO)))
(add START-OF-PIECE (fetch (PIECE PLEN) of PC))
(SETQ PC (fetch (PIECE NEXTPIECE) of PC))
```

(MBUTTON.FIND.NEXT.BUTTON

```
[LAMBDA (TEXTOBJ CH#) ; Edited 22-Apr-93 16:39 by jds
;; Finds the next instance of an OBJECT which looks like a menu button, 3-state button, or menuobj. If none is found, return NIL
(PROG ((PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
START-OF-PIECE)
(RETURN (bind PC OBJ first (SETQ PC (\CHTOPC CH# PCTB T)) while (AND PC (NOT (ATOM PC)))
do ; Loo thru the piece table, looking for pieces with objects in them
(SETQ OBJ (fetch (PIECE POBJ) of PC))
[COND
((AND OBJ (OR (type? MBUTTON OBJ)
(type? MARGINBAR OBJ)
(type? NWAYBUTTON OBJ))) ; Which are some kind of menu-buttonish object
(RETURN (CONS OBJ START-OF-PIECE)
(add START-OF-PIECE (fetch (PIECE PLEN) of PC))
(SETQ PC (fetch (PIECE NEXTPIECE) of PC))
```

(MBUTTON.FIND.NEXT.FIELD

```
[LAMBDA (TEXTOBJ CH# DON' TFIX) ; Edited 22-Apr-93 16:53 by jds
;; Starting from CH#, find the next fill-in area (usually surrounded by a {-} pair), and select any text it contains. Returns the TEXTOBJ's
;; SCRATCHSEL with the text selected. (If no insert point is found, NIL.)
(PROG ((PCTB (fetch (TEXTOBJ PCTB) of TEXTOBJ))
(SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
CH1 PCNO PCNO1 PC CH LEN START-OF-PIECE (DEPTH 0))
(COND
((IGREATERP CH# (fetch (TEXTOBJ TEXTLEN) of TEXTOBJ)) ; Can't look past the end of the document
(RETURN NIL)))
(SETQ PC (\CHTOPC CH# PCTB T)) ; Look thru the pieces for one which starts a user-fill-in area
(while PC do
(COND
((fetch (CHARLOOKS CLSELHERE) of (fetch (PIECE PLOOKS) of PC)) ; Found it, so return
(RETURN)))
(add START-OF-PIECE (fetch (PIECE PLEN) of PC))
(SETQ PC (fetch (PIECE NEXTPIECE) of PC)))
(COND
(PC ; We found a starting point for a type-in field
(add START-OF-PIECE (fetch (PIECE PLEN) of PC))
(SETQ CH1 START-OF-PIECE) ; Remember the starting character number
(SETQ PC (fetch (PIECE NEXTPIECE) of PC))
(while PC do (COND
((fetch (CHARLOOKS CLPROTECTED) of (fetch (PIECE PLOOKS) of PC))
(RETURN)))
(add START-OF-PIECE (fetch (PIECE PLEN) of PC))
(SETQ PC (fetch (PIECE NEXTPIECE) of PC)))
(SETQ LEN (IDIFFERENCE START-OF-PIECE CH1))
(replace (SELECTION CH#) of SCRATCHSEL with CH1)
(replace (SELECTION CHLIM) of SCRATCHSEL with (IPLUS CH1 (IMAX 0 LEN)))
(replace (SELECTION DCH) of SCRATCHSEL with LEN)
(replace (SELECTION SELOBJ) of SCRATCHSEL with NIL)
(replace (SELECTION POINT) of SCRATCHSEL with 'LEFT) ; So if it's used, it'll be in the correct spot.
(replace (SELECTION SELKIND) of SCRATCHSEL with 'CHAR)) ; No fill-in blank found, so return an indication.
(T
(RETURN NIL)))
(COND
((NOT DON' TFIX)
(\FIXSEL SCRATCHSEL TEXTOBJ)))
(RETURN SCRATCHSEL])
```

(MBUTTON.INIT

```
[LAMBDA NIL (* jds "12-Feb-85 14:32")
(SETQ MBUTTONIMAGEFNS (IMAGEFNSCREATE (FUNCTION MB.DISPLAY)
(FUNCTION MB.SIZEFN)
(FUNCTION MB.PUTFN)
(FUNCTION MB.GETFN)
'MB.COPYFN
(FUNCTION MB.BUTTONEVENTINFN)
'NIL
'NIL
'NIL
'NIL
'NIL
'NIL
(FUNCTION MB.WHENOPERATEDFN)
'NIL
'TEditMenuButton])
```

(MBUTTON.NEXT.FIELD.AS.NUMBER

[LAMBDA (TEXTOBJ CH#) ; Edited 12-Jun-90 19:00 by mitani
(**MBUTTON.FIND.NEXT.FIELD** TEXTOBJ CH# T)
(NUMBERP (MKATOM (TEDIT.SEL.AS.STRING (**fetch** (TEXTOBJ STREAMHINT) **of** TEXTOBJ)
(**fetch** (TEXTOBJ SCRATCHSEL) **of** TEXTOBJ]))

(MBUTTON.NEXT.FIELD.AS.PIECES

[LAMBDA (TEXTOBJ CH#) ; Edited 30-Mar-94 16:02 by jds
;; Find the next fill-in field in the menu after CH#, and return its contents as A LIST OF PIECES.
(**MBUTTON.FIND.NEXT.FIELD** TEXTOBJ CH# T)
(TEDIT.SEL.AS.PIECES TEXTOBJ (**fetch** (TEXTOBJ SCRATCHSEL) **of** TEXTOBJ)
NIL
'CL:IDENTITY])

(MBUTTON.NEXT.FIELD.AS.TEXT

[LAMBDA (TEXTOBJ CH#) ; Edited 22-Apr-93 16:14 by jds
;; Find the next fill-in field in the menu after CH#, and return its contents as a string.
(**MBUTTON.FIND.NEXT.FIELD** TEXTOBJ CH# T)
(TEDIT.SEL.AS.STRING (**fetch** (TEXTOBJ STREAMHINT) **of** TEXTOBJ)
(**fetch** (TEXTOBJ SCRATCHSEL) **of** TEXTOBJ]))

(MBUTTON.NEXT.FIELD.AS.ATOM

[LAMBDA (TEXTOBJ CH#) ; Edited 12-Jun-90 19:00 by mitani

(* Find the next fill-in field, and return its contents as an atom. If the field is empty, return NIL.)

(**MBUTTON.FIND.NEXT.FIELD** TEXTOBJ CH# T) (* Move to the next fill-in blank.)
(PROG [(STR (TEDIT.SEL.AS.STRING (**fetch** (TEXTOBJ STREAMHINT) **of** TEXTOBJ)
(**fetch** (TEXTOBJ SCRATCHSEL) **of** TEXTOBJ))
(COND
((ZEROP (NCHARS STR)) (* The field is empty.)
(RETURN NIL))
(T (* It's non-empty. Convert the string to an atom.)
(RETURN (MKATOM STR))

(MBUTTON.SET.FIELD

[LAMBDA (TEXTSTREAM FIELD VALUE) ; Edited 22-Apr-93 10:56 by jds
;; Makes the contents of the field with name FIELD be VALUE.
(PROG ((TEXTOBJ (TEXTOBJ TEXTSTREAM))
PCTB OBJ SAVED.SEL FIELD.SEL PCNO NEW-STRING)
(SETQ PCTB (**fetch** (TEXTOBJ PCTB) **of** TEXTOBJ))
(SETQ PCNO (**MBUTTON.FIND.BUTTON** FIELD TEXTSTREAM))
(COND
(PCNO [SETQ FIELD.SEL (**MBUTTON.FIND.NEXT.FIELD** TEXTOBJ (**fetch** (PCTNODE CHNUM)
of (FINDNODE-INDEX PCTB PCNO)]
; select the field following this button.
(COND
(FIELD.SEL ; there are contents to set for this button
(\FIXSEL FIELD.SEL TEXTOBJ)
(TEDIT.SETSEL TEXTSTREAM (**fetch** (SELECTION CH#) **of** FIELD.SEL)
(**fetch** (SELECTION DCH) **of** FIELD.SEL)
(**fetch** (SELECTION POINT) **of** FIELD.SEL)
T)
(SETQ NEW-STRING (MKSTRING VALUE))
(COND
((ZEROP (NCHARS NEW-STRING)) ; Nothing to replace, so just delete it.
(TEDIT.DELETE TEXTSTREAM))
(T ; there IS new info, so insert it.
(TEDIT.INSERT TEXTSTREAM (MKSTRING VALUE]))

(MBUTTON.SET.NEXT.FIELD

[LAMBDA (TEXTOBJ CH# NEWVALUE DONTUPDATESCREEN) ; Edited 30-May-91 22:15 by jds

(* SET the text content of the next fill-in field in this document to be NEWVALUE)

(PROG ((SCRATCHSEL (**fetch** (TEXTOBJ SCRATCHSEL) **of** TEXTOBJ))
(**MBUTTON.FIND.NEXT.FIELD** TEXTOBJ CH#) (* Find the next menu fill-in field)
(\FIXSEL SCRATCHSEL TEXTOBJ)

(* Fix up the SELECTION that describes its contents, so we've got the right screen coordinates &c)

(OR (ZEROP (**fetch** (SELECTION DCH) **of** SCRATCHSEL))
(\TEDIT.DELETE SCRATCHSEL TEXTOBJ T)) (* If there is text in that fill-in, delete it to make room for ours)
(COND
(NEWVALUE (* Only insert something if there IS something to insert.)
(TEDIT.\INSERT (MKSTRING NEWVALUE)
SCRATCHSEL TEXTOBJ))) (* Then fill it with out new value.)
])

(MBUTTON.SET.NEXT.BUTTON.STATE

[LAMBDA (TEXTOBJ STARTINGCH NEWSTATE)

(* jds "31-Jul-85 22:09")

(* Find the next menu button in the document, and set its state to NEWSTATE.
Return 1 + the CH# of the button, for further searchers)

(PROG* ((NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ STARTINGCH))
(BUTTON (CAR NEXTB)))
(IMAGEOBJPROP BUTTON 'STATE NEWSTATE)
(RETURN (ADD1 (CDR NEXTB]))

(TEDITMENU.STREAM

[LAMBDA (TEXTSTREAM)

(* jds "13-Aug-84 14:10")

(* returns the textstream of the teditmenu attached to this stream if any)

(PROG (MENUW (MAINWINDOW (\TEDIT.MAINW TEXTSTREAM)))
[SETQ MENUW (for w in (ATTACHEDWINDOWS MAINWINDOW) **thereis** (AND (WINDOWPROP W 'TEDITMENU)
(EQUAL (WINDOWPROP W 'TITLE)
"Tedit Menu"]

(RETURN (COND
(MENUW (TEXTSTREAM MENUW]))

(\TEDITMENU.SELSCREENER

[LAMBDA (TEXTOBJ SEL SELECTMODE FINAL?)

; Edited 30-May-91 22:15 by jds

(* Called to screen potential selections in the TEdit menu window;
if an edit op is in progress, no selection will be permitted.-)

(PROG ((MAINW (WINDOWPROP (fetch (TEXTOBJ \WINDOW) of TEXTOBJ)
'MAINWINDOW))
MAINTXT)
(SETQ MAINTXT (WINDOWPROP MAINW 'TEXTOBJ))
(COND
(AND (EQ (fetch (SELECTION CH#) of SEL)
(fetch (SELECTION CH#) of TEDIT.SCRATCHSELECTION))
(EQ (fetch (SELECTION DCH) of SEL)
(fetch (SELECTION DCH) of TEDIT.SCRATCHSELECTION))
(fetch (TEXTOBJ EDITOPACTIVE) of MAINTXT))
(\COPYSEL SEL TEDIT.SCRATCHSELECTION)
(RETURN 'DON'T))
(EQ (fetch (TEXTOBJ EDITOPACTIVE) of MAINTXT)
T)
(TEDIT.PROMPTPRINT TEXTOBJ "Edit operation in progress; please wait." T)
(RETURN 'DON'T))
(fetch (TEXTOBJ EDITOPACTIVE) of MAINTXT)
(TEDIT.PROMPTPRINT TEXTOBJ (CONCAT (fetch (TEXTOBJ EDITOPACTIVE) of MAINTXT)
" in progress; please wait."))
T)
(\COPYSEL SEL TEDIT.SCRATCHSELECTION)
(RETURN 'DON'T))

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS MBUTTONIMAGEFN)

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(MBUTTON.INIT)

(ADDTOVAR **IMAGEOBJTYPES** (TeditMenuButton FILE TEDITMENU GETFN MB.GETFN))

)

:: Three-state (ON-OFF-NEUTRAL) menu buttons, for, e.g., character properties like BOLD

(DEFINEQ

(MB.CREATE.THREESTATEBUTTON

[LAMBDA (TEXT FONT STATECHANGEFN INITSTATE)

(* jds "24-Sep-86 00:49")

(PROG ((OBJ (IMAGEOBJCREATE NIL THREESTATEIMAGEFN))
(BOX (create IMAGEBOX
XSIZE _ (STRINGWIDTH TEXT FONT)
YSIZE _ (FONTPROP FONT 'HEIGHT)
YDESC _ (FONTPROP FONT 'DESCENT)
XKERN _ 0))
DS BITMAP X Y)
(SETQ X (fetch XSIZE of BOX))
(SETQ Y (fetch YSIZE of BOX))
(IMAGEOBJPROP OBJ 'MBTEXT TEXT)
(IMAGEOBJPROP OBJ 'MBFONT FONT)
(IMAGEOBJPROP OBJ 'MBFN 'MB.THREESTATEBUTTON.FN)
(IMAGEOBJPROP OBJ 'STATECHANGEFN STATECHANGEFN)

```
(IMAGEOBJPROP OBJ 'STATE (OR INITSTATE 'NEUTRAL))
(SETQ BITMAP (BITMAPCREATE X Y))
(IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
(SETQ DS (DSPCREATE BITMAP))
(DSPXOFFSET 0 DS)
(DSPYOFFSET 0 DS)
(DSPFONT FONT DS)
(MOVETO 0 (FONTPROP FONT 'DESCENT)
  DS)
(PRIN1 (IMAGEOBJPROP OBJ 'MBTEXT)
  DS)
(RETURN OBJ)]
```

(MB.THREESTATE.DISPLAY

[LAMBDA (OBJ STREAM MODE)

(* jds "30-Aug-84 13:53")
(* Display the innards of a menu button)

```
(PROG (DS (OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX))
  (FONT (IMAGEOBJPROP OBJ 'MBFONT))
  (CURX (DSPXPOSITION NIL STREAM))
  (CURY (DSPYPOSITION NIL STREAM))
  BITMAP X Y)
(OR OBJBOX (SETQ OBJBOX (MB.SIZEFN OBJ STREAM)))
(SETQ X (fetch XSIZE of OBJBOX))
(SETQ Y (fetch YSIZE of OBJBOX))
(COND
```

(* Make sure the size is set.)

```
((SETQ BITMAP (IMAGEOBJPROP OBJ 'BITCACHE))
)
(T
```

(* The image bitmap exists already.
Use it.)

(* Need to create an image for this object.)

```
(SETQ BITMAP (BITMAPCREATE X Y))
(IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
(SETQ DS (DSPCREATE BITMAP))
(DSPXOFFSET 0 DS)
(DSPYOFFSET 0 DS)
(DSPFONT FONT DS)
(MOVETO 0 (FONTPROP FONT 'DESCENT)
  DS)
(PRIN1 (IMAGEOBJPROP OBJ 'MBTEXT)
  DS)))
(BITBLT BITMAP 0 0 STREAM CURX (IDIFFERENCE CURY (fetch YDESC of OBJBOX))
  X Y 'INPUT 'PAINT)
(SELECTQ (IMAGEOBJPROP OBJ 'STATE)
```

```
(ON
  (BITBLT NIL 0 0 STREAM CURX (IDIFFERENCE CURY (fetch YDESC of OBJBOX))
    X Y 'TEXTURE 'INVERT BLACKSHADE))
(OFF
  (DRAWLINE CURX (IDIFFERENCE CURY (fetch YDESC of OBJBOX))
    (SUB1 (IPLUS CURX X))
    (SUB1 (IPLUS (IDIFFERENCE CURY (fetch YDESC of OBJBOX))
      Y))
    1
    'PAINT STREAM))
(NEUTRAL
NIL])
```

(* The button is ON. Display it as white text on black background)

(* The button is OFF. Mark it with a diagonal line thru it.)

(* The button is neutral. Just display it regular.)

(MB.THREESTATE.SHOWSELFN

[LAMBDA (OBJ SEL ON DS)

; Edited 30-May-91 22:16 by jds

```
(PROG [(IMAGEBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
  (IMAGEBOX OBJ DS)]
```

```
(COND
  (ON (SELECTQ (IMAGEOBJPROP OBJ 'STATE)
    (ON
```

(* Switch from ON to NEUTRAL)

```
(BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
  (fetch YSIZE of IMAGEBOX)
  'TEXTURE
  'INVERT BLACKSHADE))
```

(* Switch from OFF to ON)

```
(OFF
  (BITBLT (IMAGEOBJPROP OBJ 'BITCACHE)
    0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
    (fetch YSIZE of IMAGEBOX)
    'INPUT
    'REPLACE)
  (BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
    (fetch YSIZE of IMAGEBOX)
    'TEXTURE
    'INVERT BLACKSHADE))
(NEUTRAL
```

(* Switch from NEUTRAL to OFF)

```
(DRAWLINE 0 0 (SUB1 (fetch XSIZE of IMAGEBOX))
  (SUB1 (fetch YSIZE of IMAGEBOX))
  1
  'PAINT DS))
(NIL))
((fetch (SELECTION SET) of SEL)
(SELECTQ (IMAGEOBJPROP OBJ 'STATE)
```



```

(ON (* Switch from NEUTRAL to ON)
  (BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
    (fetch YSIZE of IMAGEBOX)
    'TEXTURE
    'INVERT BLACKSHADE))
(OFF (* Switch from ON to OFF)
  (BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
    (fetch YSIZE of IMAGEBOX)
    'TEXTURE
    'INVERT BLACKSHADE)
  (DRAWLINE 0 0 (SUB1 (fetch XSIZE of IMAGEBOX))
    (SUB1 (fetch YSIZE of IMAGEBOX))
    1
    'PAINT DS))
(NEUTRAL (* Switch from OFF to NEUTRAL)
  (BITBLT (IMAGEOBJPROP OBJ 'BITCACHE)
    0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
    (fetch YSIZE of IMAGEBOX)
    'INPUT
    'REPLACE))
NIL))

```

(MB.THREESTATE.WHENOPERATEDFN

```

[LAMBDA (OBJ DS OPERATION SEL) ; Edited 30-May-91 22:16 by jds
  (* Handle operations on a three-state button)
  (SELECTQ OPERATION
    (HIGHLIGHTED (* It is being highlighted)
      (MB.THREESTATE.SHOWSELFN OBJ SEL T DS))
    (UNHIGHLIGHTED (* And being de-highlighted)
      (MB.THREESTATE.SHOWSELFN OBJ SEL NIL DS))
    (SELECTED (* It's being selected)
      (MB.THREESTATEBUTTON.FN OBJ SEL DS)
      (* Run the state-changing function)
      (* And mar the selection turned off, so others can use it without
      trashing us)
      (replace (SELECTION SET) of SEL with NIL)
      (replace (SELECTION ONFLG) of SEL with NIL)
      (replace (SELECTION SET) of TEDIT.SELECTION with NIL))
    (DESELECTED)
    NIL))

```

(MB.THREESTATEBUTTON.FN

```

[LAMBDA (OBJ SEL W) ; Edited 30-May-91 22:16 by jds
  (* MBFN for TEdit default menu item buttons.)
  (PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
    (STATECHANGEFN (IMAGEOBJPROP OBJ 'STATECHANGEFN))
    (OFF CH NEWSTATE)
    (SETQ NEWSTATE (SELECTQ (IMAGEOBJPROP OBJ 'STATE)
      (OFF 'ON)
      (ON 'NEUTRAL)
      (NEUTRAL 'OFF)
      'ON))
    (if STATECHANGEFN
      then (* apply the user supplied state change fn if she supplied one)
        (APPLY* STATECHANGEFN OBJ NEWSTATE (TEXTSTREAM TEXTOBJ)))
    (IMAGEOBJPROP OBJ 'STATE NEWSTATE)
    (replace (SELECTION ONFLG) of SEL with NIL))

```

(THREESTATE.INIT

```

[LAMBDA NIL (* jds " 9-Feb-86 15:17"
  (* Initialize the IMAGEFNS for 3-state menu button IMAGEOBJS)
  (SETQ THREESTATEIMAGEFNS (IMAGEFNSCREATE (FUNCTION MB.THREESTATE.DISPLAY)
    (FUNCTION MB.SIZEFN)
    (FUNCTION MB.PUTFN)
    (FUNCTION MB.GETFN)
    (FUNCTION MB.COPYFN)
    (FUNCTION MB.BUTTONEVENTINFN)
    'NIL
    'NIL
    'NIL
    'NIL
    'NIL
    'NIL
    (FUNCTION MB.THREESTATE.WHENOPERATEDFN)
    'NIL
    '3StateMenuButton))
)

```

(DECLARE%: DONTEVAL@LOAD DOCOPY

(THREESTATE.INIT)

)

:: One-of-N Menu button sets

(DEFINEQ

(MB.CREATE.NWAYBUTTON

```
[LAMBDA (BUTTONS FONT CHANGESTATEFN INITSTATE MAXITEMS/LINE) (* gbn "24-Sep-84 15:31")
  (PROG ((OBJECT (IMAGEOBJCREATE NIL NWAYBUTTONIMAGEFNS))
    HEIGHT IMAGES IMAGE DS DESCENT SPACING SIDEEFFECTFNS WIDTHS TWIDTHS)
    (SETQ FONT (OR FONT (FONTCREATE 'HELVETICA 10)))
    (SETQ HEIGHT (FONTPROP FONT 'HEIGHT))
    (SETQ DESCENT (FONTPROP FONT 'DESCENT))
    (SETQ WIDTHS (for BUTTON in BUTTONS collect (STRINGWIDTH (COND
      ((NLISTP BUTTON)
        BUTTON)
      (T (CAR BUTTON)))
      FONT)))
    (SETQ IMAGES (for WIDTH in WIDTHS as BUTTON in BUTTONS collect (BITMAPCREATE WIDTH HEIGHT)))
    (SETQ SPACING (STRINGWIDTH " " FONT))
    [SETQ SIDEEFFECTFNS (for BUTTON in BUTTONS collect (AND (LISTP BUTTON)
      (CADR BUTTON))
    (SETQ DS (DSPCREATE))
    (DSPXOFFSET 0 DS)
    (DSPYOFFSET 0 DS)
    (DSPFONT FONT DS)
    (DSPRIGHTMARGIN 32000 DS)
    (for IMAGE in IMAGES as BUTTON in BUTTONS do (DSPDESTINATION IMAGE DS)
      (MOVETO 0 DESCENT DS)
      (PRIN1 (COND
        ((NLISTP BUTTON)
          BUTTON)
        (T (CAR BUTTON)))
        DS))
    (IMAGEOBJPROP OBJECT 'MINWIDTH (for WIDTH in WIDTHS largest WIDTH))
    (* We always need at least one button's width)
    (IMAGEOBJPROP OBJECT 'MINHEIGHT (IPLUS HEIGHT 2)) (* And at least one button's height)
    [IMAGEOBJPROP OBJECT 'MAXWIDTH (COND
      [MAXITEMS/LINE (SETQ TWIDTHS (SORT (COPY WIDTHS)))
      (IPLUS (CAR TWIDTHS)
        (for WIDTH in (CDR TWIDTHS) as I from 1
          to (SUB1 MAXITEMS/LINE)
            sum (IPLUS WIDTH SPACING)
          (T (IPLUS (CAR WIDTHS)
            (for WIDTH in (CDR WIDTHS) sum (IPLUS WIDTH SPACING)
              (* At most, we're as wide as the N widest buttons put together)
            (LENGTH BUTTONS)))
    (IMAGEOBJPROP OBJECT 'ITEMSPACE SPACING)
    (IMAGEOBJPROP OBJECT 'BUTTONS BUTTONS)
    (IMAGEOBJPROP OBJECT 'BUTTONIMAGES IMAGES)
    (IMAGEOBJPROP OBJECT 'BUTTONHEIGHT (IPLUS HEIGHT 2))
    (IMAGEOBJPROP OBJECT 'BUTTONWIDTHS WIDTHS)
    (IMAGEOBJPROP OBJECT 'NBUTTONS (LENGTH BUTTONS))
    (IMAGEOBJPROP OBJECT 'STATE INITSTATE)
    (IMAGEOBJPROP OBJECT 'SELECTEDBUTTON NIL)
    (IMAGEOBJPROP OBJECT 'SIDEEFFECTFNS SIDEEFFECTFNS)
    (IMAGEOBJPROP OBJECT 'DESCENT DESCENT)
    (IMAGEOBJPROP OBJECT 'MBFONT FONT)
    (IMAGEOBJPROP OBJECT 'MAXITEMS/LINE MAXITEMS/LINE)
    (RETURN OBJECT])
```

(MB.NB.DISPLAYFN

```
[LAMBDA (OBJ STREAM MODE) (* jds "28-Aug-84 15:07")
  (* Display the innards of a menu button)
  (PROG (BITMAP DS (OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX))
    (X (DSPXPOSITION NIL STREAM))
    (Y (DSPYPOSITION NIL STREAM))
    (BUTTONX (IMAGEOBJPROP OBJ 'BUTTONX))
    (BUTTONY (IMAGEOBJPROP OBJ 'BUTTONY))
    (BUTTONLIST (IMAGEOBJPROP OBJ 'BUTTONS))
    (BUTTONIMAGES (IMAGEOBJPROP OBJ 'BUTTONIMAGES))
    STATE)
  [COND
    ((SETQ BITMAP (IMAGEOBJPROP OBJ 'IMAGECACHE)) (* The button image exists already)
    )
    (T (* Have to make one.)
      (SETQ BITMAP (BITMAPCREATE (fetch XSIZE of OBJBOX)
        (fetch YSIZE of OBJBOX))
      (IMAGEOBJPROP OBJ 'IMAGECACHE BITMAP)
      (SETQ DS (DSPCREATE BITMAP))
      (DSPXOFFSET 0 DS)
      (DSPYOFFSET 0 DS)
      (DSPFONT (IMAGEOBJPROP OBJ 'MBFONT)
        DS)
      (for x in BUTTONX as y in BUTTONY as image in BUTTONIMAGES
        do (* Display the images)
          (BITBLT IMAGE 0 0 DS X Y NIL NIL 'INPUT 'REPLACE]
    [BITBLT BITMAP 0 0 STREAM X (SETQ Y (IDIFFERENCE Y (fetch YDESC of OBJBOX)
      (* Display the button's image)
    (COND
```

```

((SETQ STATE (IMAGEOBJPROP OBJ 'STATE)) (* There's a selected button.)
 (for BXVAL in BUTTONX as BYVAL in BUTTNY as IMAGE in BUTTONIMAGES as BUTTN in BUTTONLIST
  when (EQ STATE BUTTN) do (BITBLT IMAGE 0 0 STREAM (IPLUS X BXVAL)
                           (IPLUS Y BYVAL)
                           NIL NIL 'INVERT 'REPLACE]))

```

(MB.NB.WHENOPERATEDFN

```

[LAMBDA (OBJ DS OPERATION SEL) ; Edited 30-May-91 22:16 by jds
 (SELECTQ OPERATION
  (HIGHLIGHTED (* (MB.SHOWSELFN OBJ SEL T DS)))
  (UNHIGHLIGHTED (* (MB.SHOWSELFN OBJ SEL NIL DS)))
 )
 (SELECTED (* There may be a side-effect to occur upon selection.)
  (PROG ((STATE (IMAGEOBJPROP OBJ 'STATE))
         FN)
        (for BUTTON in (IMAGEOBJPROP OBJ 'BUTTONS) as SIDEFN in (IMAGEOBJPROP OBJ 'SIDEFFECTFN)
         when (EQ STATE BUTTN) do (COND
          (SIDEFN (MB.SELFN OBJ SEL DS SIDEFN))
          (replace (SELECTION SET) of SEL with NIL))
        )
  (DESELECTED)
  NIL])

```

(MB.NB.SIZEFN

```

[LAMBDA (OBJ STREAM CURX RIGHTMARGIN) (* jds " 6-Sep-84 14:19"
                                       (* Tell the size of an n-way menu)

 (PROG ((OLDBOX (IMAGEOBJPROP OBJ 'BOUNDBOX))
        BOX
        (MAXITEMS/LINE (IMAGEOBJPROP OBJ 'MAXITEMS/LINE))
        (MAXWIDTH (IMAGEOBJPROP OBJ 'MAXWIDTH))
        (MINWIDTH (IMAGEOBJPROP OBJ 'MINWIDTH))
        (MAXHEIGHT (IMAGEOBJPROP OBJ 'MAXHEIGHT))
        (MINHEIGHT (IMAGEOBJPROP OBJ 'MINHEIGHT))
        (LINEHEIGHT (IMAGEOBJPROP OBJ 'LINEHEIGHT))
        (BUTTONHEIGHT (IMAGEOBJPROP OBJ 'BUTTONHEIGHT))
        (BUTTONWIDTHS (IMAGEOBJPROP OBJ 'BUTTONWIDTHS))
        (SPACING (IMAGEOBJPROP OBJ 'ITEMSPACE))
        (SLACK (IDIFFERENCE RIGHTMARGIN CURX))
        BUTTONX BUTTNY BUTTONINFO WIDTH HEIGHT)
  [COND
   ((AND (IGEQL SLACK MAXWIDTH)
         (NOT MAXITEMS/LINE))
    (* There's space for all the items on one line.
    Use it)

    (SETQ WIDTH MAXWIDTH)
    (SETQ HEIGHT MINHEIGHT)
    [SETQ BUTTONX (bind (CURX _ 0) for ITEM in BUTTONWIDTHS collect (PROG1 CURX
                                                                    (add CURX SPACING)
                                                                    (add CURX ITEM)))]

    (SETQ BUTTNY (for ITEM in BUTTONWIDTHS collect 0))
    [ (ILEQL SLACK MINWIDTH)
      (SETQ WIDTH MINWIDTH)
      (SETQ HEIGHT MAXHEIGHT)
      (SETQ BUTTONX (for ITEM in BUTTONWIDTHS collect 0))
      (SETQ BUTTNY (bind (CURY _ (ITIMES BUTTONHEIGHT (LENGTH BUTTONWIDTHS))) for ITEM in BUTTONWIDTHS
                        collect (add CURY (MINUS BUTTONHEIGHT))))
      (T (SETQ BUTTONINFO (MB.NB.PACKITEMS SLACK BUTTONWIDTHS SPACING MAXITEMS/LINE))
        [SETQ BUTTONX (for LINE in BUTTONINFO join (COPY (CDR LINE)
                                                       [SETQ BUTTNY (bind (CURY _ (ITIMES BUTTONHEIGHT (LENGTH BUTTONINFO))) for LINE in BUTTONINFO
                                                       join (PROGN (SETQ CURY (IDIFFERENCE CURY BUTTONHEIGHT))
                                                       (for X in (CDR LINE) collect CURY)]
                                                       [SETQ WIDTH (CAR (for LINE in BUTTONINFO largest (CAR LINE)
                                                       (SETQ HEIGHT (ITIMES BUTTONHEIGHT (LENGTH BUTTONINFO)

 (COND
  ((AND OLDBOX (IEQL WIDTH (fetch XSIZE of OLDBOX))
        (IEQL HEIGHT (fetch YSIZE of OLDBOX)))
   (RETURN OLDBOX))
  (T
   (IMAGEOBJPROP OBJ 'IMAGECACHE NIL)))
  (SETQ BOX (create IMAGEBOX
                   XSIZE _ WIDTH
                   YSIZE _ HEIGHT
                   YDESC _ (IMAGEOBJPROP OBJ 'DESCENT)
                   XKERN _ 0))
  (IMAGEOBJPROP OBJ 'BOUNDBOX BOX)
  (IMAGEOBJPROP OBJ 'BUTTONX BUTTONX)
  (IMAGEOBJPROP OBJ 'BUTTONY BUTTNY)
  (RETURN BOX))

```

(MB.NWAYBUTTON.SELFN

```

[LAMBDA (OBJ W SEL MOUSEX MOUSEY) ; Edited 30-May-91 22:16 by jds
                                       (* Selecting an NWAY button.)

 (PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
        (OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX))
        (OLDSTATE (IMAGEOBJPROP OBJ 'STATE))
        (BUTTONLIST (IMAGEOBJPROP OBJ 'BUTTONS))

```

```

(BUTTONX (IMAGEOBJPROP OBJ 'BUTTONX))
(BUTTONIMAGES (IMAGEOBJPROP OBJ 'BUTTONIMAGES))
(BUTTONY (IMAGEOBJPROP OBJ 'BUTTONY))
(BUTTONWIDTHS (IMAGEOBJPROP OBJ 'BUTTONWIDTHS))
(BUTTONLIST (IMAGEOBJPROP OBJ 'BUTTONLIST))
(BUTTONHEIGHT (IMAGEOBJPROP OBJ 'BUTTONHEIGHT))
CH STATE)
[for BUTTON in BUTTONLIST as X in BUTTONX as Y in BUTTONY as WIDTH in BUTTONWIDTHS as IMAGE in
BUTTONIMAGES
do (COND
  ((INSIDE? (create REGION
    LEFT _ X
    BOTTOM _ Y
    WIDTH _ WIDTH
    HEIGHT _ BUTTONHEIGHT)
    MOUSEX MOUSEY) (* The mouse is pointing here. Select this.)
  (SETQ STATE BUTTON)
  (BITBLT IMAGE 0 0 W X Y NIL NIL 'INVERT 'REPLACE))
  ((EQ OLDSTATE BUTTON) (* This was the old selection (and it's different, too)%.
Unselect it)
  (BITBLT IMAGE 0 0 W X Y NIL NIL 'INPUT 'REPLACE]
(IMAGEOBJPROP OBJ 'STATE STATE)
(RETURN T])

```

(MB.NWAYMENU.NEWBUTTON

```

[LAMBDA (TEXTOBJ CH# OLDBUTTON NEWBUTTON) (* jds "8-Feb-84 19:41")

```

(* Given a hook on an existing button, and an insertion point, insert a new button)

```

(PROG ((ARBITRATOR (IMAGEOBJPROP OLDBUTTON 'ARBITRATOR))
BUTTON)
(IMAGEOBJPROP BUTTON 'ARBITRATOR ARBITRATOR)
(TEDIT.INSERT.OBJECT BUTTON TEXTOBJ CH#)
(TEDIT.INSERT TEXTOBJ " " (ADD1 CH#))
(TEDIT.LOOKS TEXTOBJ '(PROTECTED ON)
(ADD1 CH#)
2)
(RETURN BUTTON])

```

(NWAYBUTTON.INIT

```

[LAMBDA (BUTTONS FONT INITSTATE) (* jds "9-Feb-86 15:17")

```

```

(SETQ NWAYBUTTONIMAGEFN (IMAGEFNCREATE (FUNCTION MB.NB.DISPLAYFN)
(FUNCTION MB.NB.SIZEFN)
(FUNCTION MB.PUTFN)
(FUNCTION MB.GETFN)
(FUNCTION MB.COPYFN)
(FUNCTION MB.NWAYBUTTON.SELFN)
'NIL
'NIL
'NIL
'NIL
'NIL
'NIL
(FUNCTION MB.NB.WHENOPERATEDFN)
'NIL
'NWayButton])

```

(MB.NB.PACKITEMS

```

[LAMBDA (WIDTH ITEMWIDTHS SPACING MAXITEMS/LINE) (* jds "24-Oct-84 17:42")

```

(* Pack items into lines WIDTH wide. Item widths are in ITEMWIDTHS, and each pair of items on a line is separated by SPACING. Returns a list of lists, one per line packed, of the relative X starts of the items)

```

(PROG ((CURX 0)
(LINES NIL)
(CURLINE NIL)
(CURLINEITEMS 0)
ITEM)
(while ITEMWIDTHS do (SETQ ITEM (pop ITEMWIDTHS))
(COND
  ((OR [ILESSP WIDTH (IPLUS CURX ITEM (COND
(CURLINE SPACING)
(T 0]
(AND MAXITEMS/LINE (IGEQL CURLINEITEMS MAXITEMS/LINE)))
(* Time for a new line)
(SETQ LINES (NCONC1 LINES (CONS CURX CURLINE)))
(* Add to our list of lines so far)
(SETQ CURLINE NIL)
(SETQ CURLINEITEMS 0)
(* Empty the line accumulator)
(* reset the line item count)
(SETQ CURX 0)))
(AND CURLINE (add CURX SPACING))
(SETQ CURLINE (NCONC1 CURLINE CURX))
(add CURX ITEM)
(add CURLINEITEMS 1))
[AND CURLINE (SETQ LINES (NCONC1 LINES (CONS CURX CURLINE))

```

(* Capture the last partial line, if there is one.)

(RETURN LINES)]

(MB.NWAYBUTTON.ADDITEM

[LAMBDA (OBJECT NEWBUTTON)

(* jds "11-Jul-85 12:44")

(* Given an existing n-way choice menu button, add another choice to the list)

```
(PROG ([BUTTONS (CONS NEWBUTTON (IMAGEOBJPROP OBJECT 'BUTTONS)
  HEIGHT IMAGES IMAGE DS DESCENT SPACING SIDEEFFECTFNS WIDTHS FONT)
  (SETQ FONT (IMAGEOBJPROP OBJECT 'MBFONT))
  (SETQ HEIGHT (FONTPROP FONT 'HEIGHT))
  (SETQ DESCENT (FONTPROP FONT 'DESCENT))
  (SETQ WIDTHS (for BUTTON in BUTTONS collect (STRINGWIDTH (COND
    ((NLISTP BUTTON)
     BUTTON)
    (T (CAR BUTTON))))
    FONT)))
  (SETQ IMAGES (for WIDTH in WIDTHS as BUTTON in BUTTONS collect (BITMAPCREATE WIDTH HEIGHT)))
  (SETQ SPACING (STRINGWIDTH " " FONT))
  [SETQ SIDEEFFECTFNS (for BUTTON in BUTTONS collect (AND (LISTP BUTTON)
    (CADR BUTTON))
  (SETQ DS (DSPCREATE))
  (DSPXOFFSET 0 DS)
  (DSPYOFFSET 0 DS)
  (DSPFONT FONT DS)
  (DSPRIGHTMARGIN 32000 DS)
  (for IMAGE in IMAGES as BUTTON in BUTTONS do (DSPDESTINATION IMAGE DS)
    (MOVETO 0 DESCENT DS)
    (PRIN1 (COND
      ((NLISTP BUTTON)
       BUTTON)
      (T (CAR BUTTON))))
    DS))
  (IMAGEOBJPROP OBJECT 'MINWIDTH (for WIDTH in WIDTHS largest WIDTH))
  (IMAGEOBJPROP OBJECT 'MINHEIGHT (IPLUS HEIGHT 2))
  [IMAGEOBJPROP OBJECT 'MAXWIDTH (IPLUS (CAR WIDTHS)
    (for WIDTH in (CDR WIDTHS) sum (IPLUS WIDTH SPACING)
    (LENGTH BUTTONS)))
  (IMAGEOBJPROP OBJECT 'ITEMSPACE SPACING)
  (IMAGEOBJPROP OBJECT 'BUTTONS BUTTONS)
  (IMAGEOBJPROP OBJECT 'BUTTONIMAGES IMAGES)
  (IMAGEOBJPROP OBJECT 'BUTTONHEIGHT (IPLUS HEIGHT 2))
  (IMAGEOBJPROP OBJECT 'BUTTONWIDTHS WIDTHS)
  (IMAGEOBJPROP OBJECT 'NBUTTONS (LENGTH BUTTONS))
  (IMAGEOBJPROP OBJECT 'SELECTEDBUTTON NIL)
  (IMAGEOBJPROP OBJECT 'SIDEEFFECTFNS SIDEEFFECTFNS)
  (IMAGEOBJPROP OBJECT 'DESCENT DESCENT)
  (RETURN OBJECT)])
```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS NWAYBUTTONIMAGEFNS)

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(NWAYBUTTON.INIT)

(ADDTOVAR IMAGEOBJTYPES (NWayButton FILE TEDITMENU GETFN MB.GETFN))

)

:: Two-state, toggling menu buttons.

(DEFINEQ

(TEXTMENU.TOGGLE.CREATE

[LAMBDA (TEXT FONT STATECHANGEFN INITSTATE)

(* gbn "24-Sep-84 14:45")

(* Creates a TOGGLE menu button, that can turn off and on alternately.)

```
(PROG ((OBJ (IMAGEOBJCREATE NIL \TOGGLEIMAGEFNS))
  (BOX (create IMAGEBOX
    XSIZE _ (STRINGWIDTH TEXT FONT)
    YSIZE _ (FONTPROP FONT 'HEIGHT)
    YDESC _ (FONTPROP FONT 'DESCENT)
    XKERN _ 0))
  DS BITMAP X Y)
  (SETQ X (fetch XSIZE of BOX))
  (SETQ Y (fetch YSIZE of BOX))
  (IMAGEOBJPROP OBJ 'MBTEXT TEXT)
  (IMAGEOBJPROP OBJ 'MBFONT FONT)
  (IMAGEOBJPROP OBJ 'MBFN '\TEXTMENU.TOGGLEFN)
  (IMAGEOBJPROP OBJ 'STATECHANGEFN STATECHANGEFN)
```

(* a function to be called on finalization of selection of this button to provide for user side-effects)

```
(IMAGEOBJPROP OBJ 'STATE (OR INITSTATE 'OFF))
(SETQ BITMAP (BITMAPCREATE X Y))
(IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
(SETQ DS (DSPCREATE BITMAP))
(DSPXOFFSET 0 DS)
(DSPYOFFSET 0 DS)
(DSPFONT FONT DS)
(MOVETO 0 (FONTPROP FONT 'DESCENT)
  DS)
(PRIN1 (IMAGEOBJPROP OBJ 'MBTEXT)
  DS)
(RETURN OBJ])
```

(\TEXTMENU.TOGGLE.DISPLAY

[LAMBDA (OBJ STREAM MODE)

(* gbn "27-Sep-84 01:23")
(* "27-Sep-84 01:11" gbn)
(* Display the innards of a menu toggle)

```
(PROG (DS (OBJBOX (IMAGEOBJPROP OBJ 'BOUNDBOX))
  (FONT (IMAGEOBJPROP OBJ 'MBFONT))
  (CURX (DSPXPOSITION NIL STREAM))
  (CURY (DSPYPOSITION NIL STREAM))
  BITMAP X Y)
  (OR OBJBOX (SETQ OBJBOX (MB.SIZEFN OBJ STREAM)))
  (SETQ X (fetch XSIZE of OBJBOX))
  (SETQ Y (fetch YSIZE of OBJBOX))
  (COND
```

(* Make sure the size is set.)

[[type? BITMAP (SETQ BITMAP (IMAGEOBJPROP OBJ 'BITCACHE]

(* The image bitmap exists already. Use it.)

)
(T

(* Need to create an image for this object.)

```
(SETQ BITMAP (BITMAPCREATE X Y))
(IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
(SETQ DS (DSPCREATE BITMAP))
(DSPXOFFSET 0 DS)
(DSPYOFFSET 0 DS)
(DSPFONT FONT DS)
(MOVETO 0 (FONTPROP FONT 'DESCENT)
  DS)
(PRIN1 (IMAGEOBJPROP OBJ 'MBTEXT)
  DS)))
```

(BITBLT BITMAP 0 0 STREAM CURX (IDIFFERENCE CURY (fetch YDESC of OBJBOX))
X Y 'INPUT 'PAINT)

(SELECTQ (IMAGEOBJPROP OBJ 'STATE)
(OFF

(* The button is ON. Display it as white text on black background)

(BITBLT NIL 0 0 STREAM CURX (IDIFFERENCE CURY (fetch YDESC of OBJBOX))
X Y 'TEXTURE 'INVERT BLACKSHADE))

(* The button is OFF. Just display it regular.)

(OFF
(ERROR "Invalid state in toggle button " OBJ])

(\TEXTMENU.TOGGLE.SHOWSELFN

[LAMBDA (OBJ SEL ON DS)

; Edited 30-May-91 22:16 by jds

(PROG [(IMAGEBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
 (IMAGEBOX OBJ DS)]

(COND
(OFF (SELECTQ (IMAGEOBJPROP OBJ 'STATE)
(OFF

(* Switch from ON to (NEUTRAL (* Switch from OFF to NEUTRAL) (BITBLT (IMAGEOBJPROP OBJ (QUOTE BITCACHE)) 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX) (fetch YSIZE of IMAGEBOX) (QUOTE INPUT) (QUOTE REPLACE))))

(BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
(fetch YSIZE of IMAGEBOX)
' TEXTURE
' INVERT BLACKSHADE))

(* Switch from OFF to ON)

(OFF (BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
(fetch YSIZE of IMAGEBOX)
' TEXTURE
' INVERT BLACKSHADE))
NIL))

((fetch (SELECTION SET) of SEL)
(SELECTQ (IMAGEOBJPROP OBJ 'STATE)

(* Switch from OFF to ON)

(ON (BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
(fetch YSIZE of IMAGEBOX)
' TEXTURE
' INVERT BLACKSHADE))

(* Switch from ON to OFF)

(OFF (BITBLT NIL 0 0 DS 0 0 (fetch XSIZE of IMAGEBOX)
(fetch YSIZE of IMAGEBOX)
' TEXTURE

' INVERT BLACKSHADE))

NIL])

(\TEXTMENU.TOGGLE.WHENOPERATEDFN

[LAMBDA (OBJ DS OPERATION SEL)

; Edited 30-May-91 22:16 by jds
(* Handle operations on a three-state button)

(SELECTQ OPERATION

(HIGHLIGHTED

(* It is being hilighted)

(\TEXTMENU.TOGGLE.SHOWSELFN OBJ SEL T DS))

(UNHIGHLIGHTED

(* And being de-hilighted)

(\TEXTMENU.TOGGLE.SHOWSELFN OBJ SEL NIL DS))

(SELECTED

(* It's being selected)

(\TEXTMENU.TOGGLEFN OBJ SEL DS)

(* Run the state-changing function)

(replace (SELECTION SET) of SEL with NIL)
trashing us)

(* And mar the selection turned off, so others can use it without

(replace (SELECTION ONFLG) of SEL with NIL)

(replace (SELECTION SET) of TEDIT.SELECTION with NIL))

(DESELECTED)

NIL])

(\TEXTMENU.TOGGLEFN

[LAMBDA (OBJ SEL W)

; Edited 30-May-91 22:16 by jds
(* MBFN for TOGGLE buttons--cycle back and forthe between states.)

(PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))

(STATECHANGEFN (IMAGEOBJPROP OBJ 'STATECHANGEFN))

OFFILE CH NEWSTATE)

(SETQ NEWSTATE (SELECTQ (IMAGEOBJPROP OBJ 'STATE)

(OFF 'ON)

(ON 'OFF)

'ON))

(COND

(STATECHANGEFN

(* apply the user supplied state change fn if he supplied one)

(APPLY* STATECHANGEFN OBJ NEWSTATE (TEXTSTREAM TEXTOBJ)

SEL)))

(IMAGEOBJPROP OBJ 'STATE NEWSTATE)

(replace (SELECTION ONFLG) of SEL with NIL])

(\TEXTMENU.TOGGLE.INIT

[LAMBDA NIL

(* jds "9-Feb-86 15:18")

(SETQ \TOGGLEIMAGEFNS (IMAGEFNSCREATE (FUNCTION \TEXTMENU.TOGGLE.DISPLAY)

(FUNCTION MB.SIZEFN)

(FUNCTION MB.PUTFN)

(FUNCTION MB.GETFN)

(FUNCTION MB.COPYFN)

(FUNCTION MB.BUTTONEVENTINFN)

'NIL

'NIL

'NIL

'NIL

'NIL

'NIL

(FUNCTION \TEXTMENU.TOGGLE.WHENOPERATEDFN)

'NIL

'ToggleButton])

(\TEXTMENU.SET.TOGGLE

[LAMBDA (TEXT VALUE TEXTSTREAM)

; Edited 12-Jun-90 19:02 by mitani

(* * finds the button with MBTEXT field TEXT in TEXTSTREAM and sets its state to VALUE)

(PROG ((PCNO (MBUTTON.FIND.BUTTON TEXT TEXTSTREAM))

OBJ PC)

(COND

((NOT PCNO)

(ERROR TEXT " was not found as a button.)))

[SETQ OBJ (fetch (PIECE POBJ) of (SETQ PC (fetch (PCTNODE PCE) of (FINDNODE-INDEX (fetch (TEXTOBJ PCTB)

of (TEXTOBJ

TEXTSTREAM

))

PCNO])

(IMAGEOBJPROP OBJ 'STATE VALUE)

(IMAGEOBJPROP OBJ 'BITCACHE 'JUNK)

(for WINDOW inside (fetch (TEXTOBJ \WINDOW) of (TEXTOBJ TEXTSTREAM)) do (\TEDIT.REPAINTFN WINDOW))

(RETURN VALUE])

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \TOGGLEIMAGEFNS)

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(TEXTMENU.TOGGLE.INIT)

```
(ADDOVAR IMAGEOBJTYPES (ToggleButton FILE TEDITMENU GETFN MB.GETFN))
)
```

:: Margin Setting and display

(DEFINEQ

(DRAWMARGINSCALE

```
[LAMBDA (W UNIT) ; Edited 12-Jun-90 18:59 by mitani
```

:: Draw the margin-bar scale -- the markings across the bottom of the margin bar that show you the margin values. Draws the scale in window W,
:: according to UNIT = 1 for points, or 12 for picas.

```
(PROG ((WREG (DSPCLIPPINGREGION NIL W))
      (OLDOP (DSPOPERATION 'REPLACE W)))
      (DSPFILL (create REGION
                    LEFT _ 0
                    BOTTOM _ 0
                    WIDTH _ (fetch (REGION WIDTH) of WREG)
                    HEIGHT _ 24)
            WHITESHADE
            'REPLACE W) ; CLEAR IT OUT FIRST.
      (SELECTQ UNIT
        (1 ; Straight Points
          [for X from 4 by 3 to (fetch (REGION WIDTH) of WREG)
            do ; Put a tick every 3 points, with a number every inch.
              (COND
                ((ZEROP (IREMAINDER (IDIFFERENCE X 4)
                                     72))
                 (BITBLT NIL 0 0 W X 8 1 16 'TEXTURE 'REPLACE BLACKSHADE)
                 (MOVETO (IDIFFERENCE X (LRSH (STRINGWIDTH (IDIFFERENCE X 4)
                                                1))
                          10 W)
                 (PRIN1 (IDIFFERENCE X 4)
                          W))
                 (T (BITBLT NIL 0 0 W X 20 1 4 'TEXTURE 'REPLACE BLACKSHADE])
                    ; Picas
                    (12
                     (for X from 4 by 12 to (fetch (REGION WIDTH) of WREG) as NOMX from 0
                       do ; Put a tick every half-pica, with a number every inch.
                         (COND
                           ((ZEROP (IREMAINDER NOMX 6))
                            (BITBLT NIL 0 0 W X 8 1 16 'TEXTURE 'REPLACE BLACKSHADE)
                            (MOVETO (IDIFFERENCE X (LRSH (STRINGWIDTH NOMX)
                                                           1))
                                      10 W)
                            (PRIN1 NOMX W))
                            (T (BITBLT NIL 0 0 W X 20 1 4 'TEXTURE 'REPLACE BLACKSHADE)))
                           (BITBLT NIL 0 0 W (IPLUS X 6)
                                      22 1 2 'TEXTURE 'REPLACE BLACKSHADE)))
                         NIL)
                         (BITBLT NIL 0 0 W 4 23 (fetch (REGION WIDTH) of WREG)
                          1
                          'TEXTURE
                          'REPLACE BLACKSHADE)
                         (MOVETO 0 0 W)
                         (RELDRAWTO (IDIFFERENCE (fetch (REGION WIDTH) of WREG)
                                                  2)
                                   0 1 'PAINT W)
                         (RELDRAWTO 0 (IDIFFERENCE (fetch (REGION HEIGHT) of WREG)
                                                  2)
                                   1
                                   'PAINT W)
                         (RELDRAWTO (IMINUS (IDIFFERENCE (fetch (REGION WIDTH) of WREG)
                                                         2))
                                   0 1 'PAINT W)
                         (RELDRAWTO 0 (IMINUS (IDIFFERENCE (fetch (REGION HEIGHT) of WREG)
                                                         2))
                                   1
                                   'PAINT W)
                         (DSPOPERATION OLDOP W])
        )
      )
    )
  )
)
```

(MARGINBAR

```
[LAMBDA (W L1 LN R TABS UNIT UPDATE RIGHTLIM) ; Edited 12-Jun-90 18:59 by mitani
```

(* Given a set of margins and a unit, show the margin bar properly)

```
(PROG ((OLDOP (DSPOPERATION 'ERASE W))
      (SCALEDL1 (MSCALE L1 UNIT))
      (SCALEDLN (MSCALE LN UNIT))
      (SCALEDR (MSCALE R UNIT))
      (FLOATINGRIGHT NIL)
      (EXTENDEDRIGHT NIL)
      UNSETL1 UNSETLN)
```



```

(OR UPDATE (DRAWMARGINSCALE W UNIT))
(DSPFONT (FONTCREATE 'GACHA 10)
  W)
(SETQ L1 (MKSTRING (ABS L1)))
(SETQ LN (MKSTRING (ABS LN)))
(SETQ R (MKSTRING (ABS R)))
[COND
  [(ILESSP SCALEDL1 4)
    (SETQ FLOATINGRIGHT T)
    (SETQ SCALEDL1 (IPLUS 4 (IDIFFERENCE 4 SCALEDL1)))
    ((ILEQ SCALEDL1 4)
      (SETQ FLOATINGRIGHT T)
      (SETQ SCALEDL1 RIGHTLIM))
    ((IGREATERP SCALEDL1 RIGHTLIM)
      (SETQ EXTENDEDRIGHT T)
      (SETQ SCALEDL1 (IDIFFERENCE RIGHTLIM 8)])
  ]
[COND
  ((ILESSP SCALEDL1 4)
    (SETQ UNSETL1 T)
    (SETQ SCALEDL1 (IPLUS 4 (IDIFFERENCE 4 SCALEDL1)))
  ]
[COND
  ((ILESSP SCALEDLN 4)
    (SETQ UNSETLN T)
    (SETQ SCALEDLN (IPLUS 4 (IDIFFERENCE 4 SCALEDLN)))
  ]
(BITBLT NIL 0 0 W 1 26 (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL W))
  3)
  32
  'TEXTURE
  'REPLACE WHITESHADE)
(BITBLT NIL 0 0 W SCALEDL1 42 (IDIFFERENCE SCALEDL1 SCALEDL1)
  16
  'TEXTURE
  'REPLACE BLACKSHADE)
(BITBLT NIL 0 0 W SCALEDLN 26 (IDIFFERENCE SCALEDL1 SCALEDLN)
  16
  'TEXTURE
  'REPLACE BLACKSHADE)
(COND
  (UNSETL1
    (BITBLT NIL 0 0 W SCALEDL1 42 (IPLUS (STRINGWIDTH L1 W)
      2)
      16
      'TEXTURE
      'REPLACE EDITGRAY)
    (DSPOPERATION 'PAINT W)
    (MOVETO (IPLUS SCALEDL1 2)
      44 W)
    (PRIN1 L1 W)
    (DSPOPERATION 'ERASE W))
  (T (MOVETO (IPLUS SCALEDL1 2)
    44 W)
    (PRIN1 L1 W)))
(COND
  (UNSETLN
    (BITBLT NIL 0 0 W SCALEDLN 26 (IPLUS (STRINGWIDTH LN W)
      2)
      16
      'TEXTURE
      'REPLACE EDITGRAY)
    (DSPOPERATION 'PAINT W)
    (MOVETO (IPLUS SCALEDLN 2)
      28 W)
    (PRIN1 LN W)
    (DSPOPERATION 'ERASE W))
  (T (MOVETO (IPLUS SCALEDLN 2)
    28 W)
    (PRIN1 LN W)))
[COND
  (FLOATINGRIGHT
    (BITBLT NIL 0 0 W (IDIFFERENCE SCALEDL1 (IPLUS (STRINGWIDTH R W)
      2))
      26
      (IPLUS (STRINGWIDTH R W)
        2)
      32
      'TEXTURE
      'REPLACE EDITGRAY)
    (DSPOPERATION 'PAINT W))
  (EXTENDEDRIGHT
    (BITBLT TEDIT.EXTENDEDRIGHTMARK 0 0 W SCALEDL1 26 8 32 'INPUT 'REPLACE]
(MOVETO (IDIFFERENCE SCALEDL1 (IPLUS (STRINGWIDTH R W)
  2))
  36 W)

```

(* Unset right margin. Show specially, but at its usual place.)

(* Floating right margin => marked specially)

(* Not floating, so just limit it to the rightmost that can be seen.)

(* Unset right FIRST LEFT margin. Show specially, but at its usual place.)

(* Unset LEFT margin. Show specially, but at its usual place.)

(* 1st left margin isn't set, tho it has a value. Mark it neutral)

(* left margin isn't set, tho it has a value. Mark it neutral)

(* Floating right margin is marked by a light gray marker)

(* A non-visible right margin is marked by two wavy lines indicating a break)

```

(PRIN1 R W)
(DSOPERATION OLDOP W)
(COND
  (EQ TABS 'NEUTRAL)
  (DSPFILL (create REGION
            LEFT _ 2
            BOTTOM _ 1
            HEIGHT _ 8
            WIDTH _ (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL W))
                              4))
            EDITGRAY
            'REPLACE W))
  (T (DSPFILL (create REGION
              LEFT _ 2
              BOTTOM _ 1
              HEIGHT _ 8
              WIDTH _ (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL W))
                                4))
              WHITESHADE
              'REPLACE W)
      (for TAB in TABS do
        (MB.MARGINBAR.SHOWTAB W TAB UNIT 'PAINT]))

```

(* All tabs have been neutralized. Just lay down a grey pattern over them.)

(* Run thru the tabs, putting them down in place.)

(MARGINBAR.CREATE

```
[LAMBDA (MARL1 MARLN MARR MARTABS MARUNIT MARTABTYPE)
```

; Edited 12-Jun-90 18:59 by mitani
(* Create an instance of the margin-setting ruler for TEdit's use.)

```

(PROG ((BOX (create IMAGEBOX
                  XSIZE _ 1008
                  YSIZE _ 62
                  YDESC _ 0
                  XKERN _ 4))
       (OBJ OBJDATUM BITMAP DS)
       (SETQ OBJ
         (IMAGEOBJCREATE (SETQ OBJDATUM
                              (create MARGINBAR
                                      MARL1 _ MARL1
                                      MARLN _ MARLN
                                      MARR _ MARR
                                      MARTABS _ MARTABS
                                      MARUNIT _ MARUNIT
                                      MARTABTYPE _ MARTABTYPE))
                          MARGINBARIMAGEFNS))

```

(* Create an IMAGEOBJ, containing an instance of the record to hold margin and tab info)

```

(SETQ BITMAP (BITMAPCREATE (fetch XSIZE of BOX)
                          (fetch YSIZE of BOX)))
(IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
(SETQ DS (DSPCREATE BITMAP))
(IMAGEOBJPROP OBJ 'DSPCACHE DS)
(DSPXOFFSET 0 DS)
(DSPYOFFSET 0 DS)
(DSPCLIPPINGREGION (create REGION
                          LEFT _ 0
                          BOTTOM _ 0
                          WIDTH _ (fetch XSIZE of BOX)
                          HEIGHT _ (fetch YSIZE of BOX))
                   DS)
(MARGINBAR DS (fetch (MARGINBAR MARL1) of OBJDATUM)
             (fetch (MARGINBAR MARLN) of OBJDATUM)
             (fetch (MARGINBAR MARR) of OBJDATUM)
             (fetch (MARGINBAR MARTABS) of OBJDATUM)
             (fetch (MARGINBAR MARUNIT) of OBJDATUM)
             NIL
             (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL DS)))

```

(* A cache for the ruler's screen image)

(* And a displaystream for modifying that image)

(* Fill in the cache with the original value This does the time-consuming part of drawing the ticks on the ruler and such, which would make drawing it on the fly unbearable.)

```
(IMAGEOBJPROP OBJ 'NEEDSUPDATE T)
```

(* And tell the display function that it needs to be updated when first displayed. Which is the faster part.)

```
(RETURN OBJ)]
```

(MB.MARGINBAR.SELFN

```
[LAMBDA (OBJ SELWINDOW SEL RELX RELY STREAM ORIGX ORIGY)
```

; Edited 12-Jun-90 18:59 by mitani
; Let the user adjust margins and tabs using the mouse.

```

(PROG [(OBJDATUM (IMAGEOBJPROP OBJ 'OBJECTDATUM))
      (IMAGEBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
                   (IMAGEBOX OBJ STREAM 'DISPLAY))
      (PROG ((L1 (fetch MARL1 of OBJDATUM))
            (LN (fetch MARLN of OBJDATUM))

```

```

(R (fetch MARR of OBJDATUM))
(TABS (fetch MARTABS of OBJDATUM))
[SCALED TABS (COND
  ((LISTP (fetch MARTABS of OBJDATUM))
    ; Only scale the tabs if there are any, and they're not neutralized.
    (for TAB in (fetch MARTABS of OBJDATUM) collect (MSCALE (fetch TABX of TAB)
      (fetch MARUNIT of OBJDATUM))
    )
  )
)
(UNIT (fetch MARUNIT of OBJDATUM))
(CLIP (create REGION
  LEFT _ 0
  BOTTOM _ 0
  WIDTH _ (fetch XSIZE of IMAGEBOX)
  HEIGHT _ (fetch YSIZE of IMAGEBOX))
)
(RIGHTLIM (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL SELWINDOW))
  4))
TAB TABX OLI OLN OR)
(SETQ OLI L1)
(SETQ OLN LN)
(SETQ OR R)
[COND
  [(INSIDE? (create REGION
    LEFT _ (IDIFFERENCE (MSCALE (ABS L1)
      UNIT)
    2)
    BOTTOM _ 42
    WIDTH _ 16
    HEIGHT _ 16)
    RELX RELY) ; Move the 1st-line left margin.
  (while (AND (MOUSESTATE (OR LEFT MIDDLE RIGHT))
    (INSIDE? CLIP (LASTMOUSEX STREAM)
      (LASTMOUSEY STREAM)))
    do (SETQ L1 (MAX 0 (MDESCALE (LASTMOUSEX STREAM)
      UNIT)))
    [COND
      ((\TEDIT.MOUSESTATE RIGHT) ; Right mouse button UNsets the margin.
        (SETQ L1 (MINUS L1))
      )
    ]
    (COND
      ((NOT (EQUAL OLI L1))
        (MARGINBAR STREAM L1 LN R TABS UNIT T RIGHTLIM)
        (SETQ OLI L1)
      )
    ]
  )
  [(INSIDE? (create REGION
    LEFT _ (IDIFFERENCE (MSCALE (ABS LN)
      UNIT)
    2)
    BOTTOM _ 26
    WIDTH _ 16
    HEIGHT _ 16)
    RELX RELY) ; Move the skirt's left margin
  (while (AND (MOUSESTATE (OR LEFT MIDDLE RIGHT))
    (INSIDE? CLIP (LASTMOUSEX STREAM)
      (LASTMOUSEY STREAM)))
    do (SETQ LN (MAX 0 (MDESCALE (LASTMOUSEX STREAM)
      UNIT)))
    [COND
      ((\TEDIT.MOUSESTATE RIGHT) ; Right mouse button UNsets the margin.
        (SETQ LN (MINUS LN))
      )
    ]
    (COND
      ((NOT (EQUAL OLN LN))
        (MARGINBAR STREAM L1 LN R TABS UNIT T RIGHTLIM)
        (SETQ OLN LN)
      )
    ]
  )
  [(OR (INSIDE? (create REGION
    LEFT _ (IDIFFERENCE (IMIN (MSCALE (ABS R)
      UNIT)
      (fetch XSIZE of IMAGEBOX)
      (fetch (REGION WIDTH) of (DSPCLIPPINGREGION
        NIL SELWINDOW)))
    16)
    BOTTOM _ 26
    WIDTH _ 16
    HEIGHT _ 32)
    RELX RELY)
  (AND (ZEROP (IABS (FIXR R)))
    (INSIDE? (create REGION
      LEFT _ (IDIFFERENCE (IMIN (fetch XSIZE of IMAGEBOX)
        (fetch (REGION WIDTH)
          of (DSPCLIPPINGREGION NIL SELWINDOW)))
      16)
      BOTTOM _ 26
      WIDTH _ 16
      HEIGHT _ 32)
      RELX RELY)) ; Move the right margin
  (while (AND (MOUSESTATE (OR LEFT MIDDLE RIGHT))
    (INSIDE? CLIP (LASTMOUSEX STREAM)
      (LASTMOUSEY STREAM)))
    do (SETQ R (MAX 0 (MDESCALE (LASTMOUSEX STREAM)
      UNIT)))
    [COND
  ]

```

```
(\TEDIT.MOUSESTATE RIGHT) ; Right mouse button UNsets the margin.
(SETQ R (MINUS R])
(COND
  ((NOT (EQUAL OR R))
    (MARGINBAR STREAM L1 LN R TABS UNIT T RIGHTLIM)
    (SETQ OR R]
  ((INSIDE? (create REGION
    LEFT _ 0
    BOTTOM _ 0
    WIDTH _ (fetch (REGION WIDTH) of CLIP)
    HEIGHT _ 16)
    RELX RELY) ; We're in the tab ruler region
  (COND
    ((MOUSESTATE LEFT) ; MOVE a tab
      [SETQ TAB (for TABX in SCALEDTABS as TAB in TABS smallest (ABS (IDIFFERENCE TABX
        (LASTMOUSEX STREAM)
      (AND TAB (MB.MARGINBAR.TABTRACK STREAM OBJDATUM TAB)))
      [(MOUSESTATE MIDDLE) ; ADD/CHANGE a tab
      (COND
        ((EQ (fetch MARTABS of OBJDATUM)
          'NEUTRAL) ; The tabs used to be NEUTRAL. Clear the tab region, and start
          ; afresh.
          (replace MARTABS of OBJDATUM with NIL) ; So we don't come this way again.
          (DSPFILL (create REGION
            LEFT _ 2
            BOTTOM _ 1
            HEIGHT _ 8
            WIDTH _ (IDIFFERENCE (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL
              SELWINDOW))
            4))
            WHITESHADE
            'REPLACE SELWINDOW) ; Make the tab region look non-neutral, too, so that tabs look OK
            ; on it.
          ))
        (COND
          ((AND [SETQ TAB (for TABX in SCALEDTABS as TAB in TABS
            smallest (ABS (IDIFFERENCE TABX (LASTMOUSEX STREAM)
            (SETQ TABX (MSCALE (CAR TAB)
              UNIT))
            (IGEQ (LASTMOUSEX STREAM)
              (IDIFFERENCE TABX 2))
            (ILEQ (LASTMOUSEX STREAM)
              (IPLUS TABX 2)))
            (MB.MARGINBAR.SHOWTAB STREAM TAB UNIT 'ERASE)
            (replace TABKIND of TAB with (OR (fetch MARTABTYPE of OBJDATUM)
              'LEFT))
            (MB.MARGINBAR.SHOWTAB STREAM TAB UNIT 'PAINT)
            (MB.MARGINBAR.TABTRACK STREAM OBJDATUM TAB))
            [(OR (NOT TAB)
              (NOT (EQP (fetch TABX of TAB)
                (MDESCALE (LASTMOUSEX STREAM)
                  UNIT] ; Really create a new tab
            [SETQ TAB (create TAB
              TABX _ (MDESCALE (LASTMOUSEX STREAM)
                UNIT)
              TABKIND _ (OR (fetch MARTABTYPE of OBJDATUM)
                'LEFT]
            (SETQ TABS (CONS TAB TABS))
            (MB.MARGINBAR.SHOWTAB STREAM TAB UNIT 'PAINT)
            (MB.MARGINBAR.TABTRACK STREAM OBJDATUM TAB]
            ((MOUSESTATE RIGHT) ; DELETE a tab.
            (COND
              ((AND [SETQ TAB (for TABX in SCALEDTABS as TAB in TABS
                smallest (ABS (IDIFFERENCE TABX (LASTMOUSEX STREAM)
                (SETQ TABX (MSCALE (CAR TAB)
                  UNIT))
                (IGEQ (LASTMOUSEX STREAM)
                  (IDIFFERENCE TABX 2))
                (ILEQ (LASTMOUSEX STREAM)
                  (IPLUS TABX 2)))
                (MB.MARGINBAR.SHOWTAB STREAM TAB UNIT 'ERASE)
                (SETQ TABS (LDIFFERENCE TABS (LIST TAB]
            (replace MARL1 of OBJDATUM with L1)
            (replace MARLN of OBJDATUM with LN)
            (replace MARR of OBJDATUM with R)
            (replace MARTABS of OBJDATUM with TABS)))
    T]))
```

(MB.MARGINBAR.SIZEFN

```
[LAMBDA (OBJ)
  (PROG ((BOX (create IMAGEBOX
    XSIZE _ 1008
    YSIZE _ 62
    YDESC _ 0
    XKERN _ 4)))
```

(* jds " 5-Sep-84 14:10")

```
(IMAGEOBJPROP OBJ 'BOUNDBOX BOX)
(RETURN BOX])
```

(MB.MARGINBAR.DISPLAYFN

[LAMBDA (OBJ STREAM MODE)

; Edited 12-Jun-90 18:59 by mitani
 (* Display the innards of a menu button)

```
(PROG ((IMAGEBOX (OR (IMAGEOBJPROP OBJ 'BOUNDBOX)
                     (IMAGEBOX OBJ STREAM MODE)))
        (OBJDATUM (IMAGEOBJPROP OBJ 'OBJECTDATUM))
        BITMAP
        (DS (DSPCREATE))
        WASON)
(COND
 [[SETQ WASON (SETQ BITMAP (IMAGEOBJPROP OBJ 'BITCACHE]
```

(* The marginbar existed already as an image. Don't bother re-creating it, and remember that we're allowed to MODIFY the old image instead of creating a new one.)

```
(SETQ DS (IMAGEOBJPROP OBJ 'DSPCACHE]
(T
 (SETQ BITMAP (BITMAPCREATE (fetch XSIZE of IMAGEBOX)
                             (fetch YSIZE of IMAGEBOX))) (* Create a cache bitmap)
 (IMAGEOBJPROP OBJ 'BITCACHE BITMAP)
 (SETQ DS (DSPCREATE BITMAP))
 (IMAGEOBJPROP OBJ 'DSPCACHE DS)
 (DSPXOFFSET 0 DS)
 (DSPYOFFSET 0 DS)
 (DSPCLIPPINGREGION (create REGION
                            LEFT _ 0
                            BOTTOM _ 0
                            WIDTH _ (fetch XSIZE of IMAGEBOX)
                            HEIGHT _ (fetch YSIZE of IMAGEBOX))
                     DS)))
(MARGINBAR DS (fetch (MARGINBAR MARL1) of OBJDATUM)
             (fetch (MARGINBAR MARLN) of OBJDATUM)
             (fetch (MARGINBAR MARR) of OBJDATUM)
             (fetch (MARGINBAR MARTABS) of OBJDATUM)
             (fetch (MARGINBAR MARUNIT) of OBJDATUM)
             (OR WASON (IMAGEOBJPROP OBJ 'NEEDSUPDATE NIL))
             (fetch (REGION WIDTH) of (DSPCLIPPINGREGION NIL STREAM)))
(* Update the image, if it needs it)
(BITBLT BITMAP 0 0 STREAM (IDIFFERENCE (DSPXPOSITION NIL STREAM)
                                         4)
        (IDIFFERENCE (DSPYPOSITION NIL STREAM)
                     (fetch YDESC of IMAGEBOX])
```

(MDESCALE

[LAMBDA (VAL UNIT)

(* jds " 4-NOV-83 17:29")
 (* Convert a value from screen offset units to marginbar units)

```
(COND
 ((IEQP UNIT 12)
  (QUOTIENT (IQUOTIENT (LLSH (IDIFFERENCE VAL 4)
                             1)
             UNIT)
            2.0))
 (T (QUOTIENT (DIFFERENCE VAL 4)
              UNIT]))
```

(MSCALE

[LAMBDA (VAL UNIT)

(* jds " 4-NOV-83 17:31")
 (* Convert from marginbar units to a screen X offset)

```
(IPLUS 4 (FIXR (TIMES VAL (OR UNIT 1]))
```

(MB.MARGINBAR.SHOWTAB

[LAMBDA (W TAB UNIT MODE)

(* jds "22-Mar-85 17:36")
 (* Paint/erase/otherwise display the sign for a TAB in window WINDOW, using units UNIT)

```
(PROG ((TABX (MSCALE (fetch TABX of TAB)
                    UNIT)))
        (SELECTQ (fetch TABKIND of TAB)
        (LEFT
         (BITBLT \TEDIT.LEFTTAB 0 0 W (IDIFFERENCE TABX 2)
                 1 NIL NIL 'INPUT MODE)) (* Flush-left tab.)
        (CENTERED
         (BITBLT \TEDIT.CENTERTAB 0 0 W (IDIFFERENCE TABX 5)
                 1 NIL NIL 'INPUT MODE)) (* Centered Tab)
        (RIGHT
         (BITBLT \TEDIT.RIGHTTAB 0 0 W (IDIFFERENCE TABX 7)
                 1 NIL NIL 'INPUT MODE)) (* Flush-right Tab)
        (DECIMAL
         (BITBLT \TEDIT.DECIMALTAB 0 0 W (IDIFFERENCE TABX 7)
                 1 NIL NIL 'INPUT MODE)) (* Decimal aligned tab)
        (DOTTEDLEFT
         (BITBLT \TEDIT.DECIMALTAB 0 0 W (IDIFFERENCE TABX 7)
                 1 NIL NIL 'INPUT MODE)) (* Decimal aligned tab)
```



```
)
(RPAQQ \TEDIT.LEFTTAB ⌞ )
(RPAQQ \TEDIT.CENTERTAB ⌞ )
(RPAQQ \TEDIT.RIGHTTAB ⌞ )
(RPAQQ \TEDIT.DECIMALTAB ⌞ )
(RPAQQ \TEDIT.DOTTED.LEFTTAB ..⌞ )
(RPAQQ \TEDIT.DOTTED.CENTERTAB ..⌞ )
(RPAQQ \TEDIT.DOTTED.RIGHTTAB ..⌞ )
(RPAQQ \TEDIT.DOTTED.DECIMALTAB ..⌞ )
(RPAQQ TEDIT.EXTENDEDRIGHTMARK ⌞⌞⌞)
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS MARGINBARIMAGEFNS
)
(DECLARE%: DONTEVAL@LOAD DOCOPY
```

(MARGINBAR.INIT)

```
(ADDTOVAR IMAGEOBJTYPES (MarginRuler FILE TEDITMENU GETFN MB.GETFN)
)
```

:: Text menu creation and support

(DEFINEQ

(\TEXTMENU.START

```
[LAMBDA (MENU MAINWINDOW TITLE HEIGHT) ; Edited 4-Jun-93 11:59 by sybalsky:mv:envos
```

:: Create a TEdit-based menu for a given main window.

```
(PROG ([WREG (COND
(MAINWINDOW (WINDOWPROP MAINWINDOW 'REGION))
(T (GETREGION)
(CH#1 NIL)
MENUW MENUTEXT)
(COND
((AND MAINWINDOW (WINDOWPROP MAINWINDOW 'TEDITMENU))
(RETURN)) ; This is a menu window. It can't have a menu, so bail out.
([AND MAINWINDOW (for ww in (ATTACHEDWINDOWS MAINWINDOW) thereis (EQUAL (OR TITLE "TEdit Menu")
(WINDOWPROP WW 'TEDITMENU))
; If this main window already has a menu, don't add another.
(RETURN)))
(SETQ MENUW (CREATEW (SETQ WREG (COND
```

```
(MAINWINDOW (create REGION
LEFT _ (fetch (REGION LEFT) of WREG)
BOTTOM _ (fetch (REGION TOP) of WREG)
WIDTH _ (fetch (REGION WIDTH) of WREG)
HEIGHT _ (OR HEIGHT 133)))
(T WREG)))
```

```
(OR TITLE "TEdit Menu")))
(WINDOWADDPROP MENUW 'CLOSEFN 'TEXTMENU.CLOSEFN)
(WINDOWPROP MENUW 'TEDITMENU (OR TITLE "TEdit Menu")) ; Mark this as a TEDIT MENU window
(ATTACHWINDOW MENUW MAINWINDOW 'TOP 'JUSTIFY 'LOCALCLOSE)
(SETQ MENUTEXT MENUW)
(replace (TEXTOBJ MENUW) of (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT) with T)
[AND MAINWINDOW (WINDOWPROP MENUW 'PROMPTWINDOW (WINDOWPROP MAINWINDOW 'PROMPTWINDOW)
[TEDIT MENUTEXT MENUW NIL (LIST 'TITLEMENUFN 'DON'T 'PROMPTWINDOW (fetch (TEXTOBJ PROMPTWINDOW)
of (TEXTOBJ MAINWINDOW))
(AND MAINWINDOW (TTY.PROCESS (WINDOWPROP MAINWINDOW 'PROCESS))]
```

(\TEXTMENU.DOC.CREATE

```
[LAMBDA (MENEDESC MENUPROPS) ; Edited 12-Jun-90 19:00 by mitani
```

(* Create the TEXTSTREAM for a menu, given a description. That stream is passed to \TEXTMENU.START to get the menu up on screen)

```
(PROG ((CH#1 NIL)
MENUW MENUTEXT)
```

```

[SETQ MENUTEXT (OPENTEXTSTREAM "" NIL NIL NIL (OR MENUPROPS '(FONT (MODERN 10]
(bind (CH# _ 1)
  OBJ for DESC in MENUDESC do (SELECTQ (CAR DESC)
    (*
      (* This is a comment within a menu description --
      ignore it.))
      (* A menu button -- hitting it calls a function)
      (MB.BUTTON
        (TEDIT.INSERT.OBJECT (MB.BUTTON.CREATE
          (MKATOM (fetch (MB.BUTTON MBLABEL
            )
              of DESC))
            (fetch (MB.BUTTON MBBUTTONEVENTFN
              )
                of DESC)
            (fetch (MB.BUTTON MBFONT)
              of DESC))
          MENUTEXT CH#)
        (TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
          '(PROTECTED OFF)
          CH# 1)
        (add CH# 1))
      (* 3-state button; hitting it changes state among ON, OFF, and
      NEUTRAL.)
      (MB.3STATE
        (TEDIT.INSERT.OBJECT (MB.CREATE.THREESTATEBUTTON
          (MKATOM (fetch (MB.3STATE MBLABEL
            )
              of DESC))
            (fetch (MB.3STATE MBFONT)
              of DESC)
            (fetch (MB.3STATE MBCHANGESTATEFN
              )
                of DESC)
            (fetch (MB.3STATE MBINITSTATE)
              of DESC))
          MENUTEXT CH#)
        (TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
          '(PROTECTED OFF)
          CH# 1)
        (add CH# 1))
      (* TOGGLE button; hitting it switches between ON and OFF.)
      (MB.TOGGLE
        (TEDIT.INSERT.OBJECT (\TEXTMENU.TOGGLE.CREATE
          (MKATOM (fetch (MB.TOGGLE MBTEXT)
            of DESC))
            (fetch (MB.TOGGLE MBFONT)
              of DESC)
            (fetch (MB.TOGGLE MBCHANGESTATEFN
              )
                of DESC)
            (fetch (MB.TOGGLE MBINITSTATE)
              of DESC))
          MENUTEXT CH#)
        (TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
          '(PROTECTED OFF)
          CH# 1)
        (add CH# 1))
      (* N-way buttons; choosing one turns the others off.)
      (MB.NWAY
        (SETQ OBJ (MB.CREATE.NWAYBUTTON (fetch (MB.NWAY
          MBBUTTONS
            )
              of DESC)
            (fetch (MB.NWAY MBFONT) of DESC)
            (fetch (MB.NWAY MBCHANGESTATEFN)
              of DESC)
            (fetch (MB.NWAY MBINITSTATE)
              of DESC)
            (fetch (MB.NWAY MBMAXITEMSPERLINE)
              of DESC)))
          (TEDIT.INSERT.OBJECT OBJ MENUTEXT CH#)
          (TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
            '(PROTECTED OFF)
            CH# 1)
          (add CH# 1))
      (* Real menu, except the selection sticks)
      (MENU
        (TEDIT.INSERT.OBJECT (MB.CREATE.FULLMENU (CADR DESC)
          MENUTEXT CH#)
          (TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
            '(PROTECTED OFF)
            CH# 1)
          (add CH# 1))
      (* Margin ruler for TEdit formatting)
      (MB.MARGINBAR
        (TEDIT.INSERT.OBJECT (MARGINBAR.CREATE -0.5 -0.5
          -39.5 NIL 12)
          MENUTEXT CH#)
        (TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ)
          of MENUTEXT)
          '(PROTECTED OFF)
          CH# 1)
        (add CH# 1))

```



```
(MB.TEXT (* Arbitrary text, which will be protected from the user.)
(TEDIT.INSERT MENUTEXT (fetch (MB.TEXT MBSTRING)
of DESC)
CH#)
[AND (fetch (MB.TEXT MBFONT) of DESC)
(TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ)
of MENUTEXT)
(LIST 'MBFONT (fetch (MB.TEXT MBFONT)
of DESC))
CH#
(NCHARS (fetch (MB.TEXT MBSTRING)
of DESC)
(TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
'(PROTECTED ON)
CH#
(NCHARS (fetch (MB.TEXT MBSTRING) of DESC)))
(add CH# (NCHARS (fetch (MB.TEXT MBSTRING) of DESC))))
(MB.INSERT (* An insertion point, with optional text to put there)
(TEDIT.INSERT MENUTEXT " {}" CH#)
(TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
'(PROTECTED ON)
CH# 4)
(TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT)
'(PROTECTED ON SELECTPOINT ON)
(IPLUS CH# 2)
1)
(OR CH#1 (SETQ CH#1 (IPLUS CH# 3)))
[COND
((fetch (MB.INSERT MBINITENTRY) of DESC)
(* There is an initial entry to be made.
Make it)
[COND
((IMAGEOBJP (fetch (MB.INSERT MBINITENTRY)
of DESC))
(* It is an imageobj.)
(TEDIT.INSERT.OBJECT (fetch (MB.INSERT
MBINITENTRY)
of DESC)
MENUTEXT
(IPLUS CH# 3)))
(T
(* It's regular text.)
(TEDIT.INSERT MENUTEXT
(MKSTRING (fetch (MB.INSERT
MBINITENTRY)
of DESC))
(IPLUS CH# 3)
[TEDIT.LOOKS (fetch (TEXTSTREAM TEXTOBJ)
of MENUTEXT)
'(PROTECTED OFF SELECTPOINT OFF)
(IPLUS CH# 3)
(NCHARS (MKSTRING (fetch (MB.INSERT
MBINITENTRY)
of DESC)
(add CH# (NCHARS (fetch (MB.INSERT MBINITENTRY)
of DESC)
(add CH# 4))
(\ILLEGAL.ARG DESC)))
(replace (TEXTOBJ MENUFLG) of (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT) with T)
(* Remember that this is a menu)
[COND
(CH#1 (* We actually inserted some text, so it makes sense to put up a
selection)
(push (fetch (TEXTOBJ EDITPROPS) of (fetch (TEXTSTREAM TEXTOBJ) of MENUTEXT))
(LIST 'SEL CH#1)
(* And where the first selection should be.)
(RETURN MENUTEXT)])
```

(TEXTMENU.CLOSEFN

[LAMBDA (W)

; Edited 12-Jun-90 18:59 by mitani

:: CLOSE a TEdit menu window: Detach the menu, then reshape the remaining windows to take up the remaining space

```
(PROG ((MAINW (WINDOWPROP W 'MAINWINDOW))
TEXTOBJ HEIGHT OHEIGHT OBOTTOM WBOTTOM WINDOWS)
(FREEATTACHEDWINDOW W)
```

```
(* (DETACHWINDOW W) (* ; "So detach this window.")
(COND ((IGREATERP (FLENGTH
(ATTACHEDWINDOWS MAINW)) 1)
(SETQ OHEIGHT (fetch (REGION HEIGHT) of
(WINDOWPROP W (QUOTE REGION))))
(SETQ OBOTTOM (fetch (REGION BOTTOM) of
(WINDOWPROP W (QUOTE REGION))))
(CLOSEW W) (SETQ WINDOWS
(SORT (ATTACHEDWINDOWS MAINW)
(FUNCTION (LAMBDA (WW) (fetch
(REGION BOTTOM) of (WINDOWPROP WW
(QUOTE REGION)))))) (for WW in WINDOWS when
(IGEQ (SETQ WBOTTOM (fetch
(REGION BOTTOM) of (WINDOWPROP WW
```

(QUOTE REGION))) OBOTTOM) do
(MOVEW WW (fetch (REGION LEFT) of
(WINDOWPROP WW (QUOTE REGION)))
(IDIFFERENCE WBOTTOM OHEIGHT))))))

(COND
((SETQ TEXTOBJ (WINDOWPROP W 'TEXTOBJ))
(TEDIT.KILL TEXTOBJ)

; Then, if this window still has a textobj under it, kill off that edit
; process.

;; This has to be TEDIT.KILL to avoid problems with the TTY being handed from main back to menu, causing main never to finish off;
;; menu would quit and hand TTY to top level window.

])

(RPAQQ TEXTMENUICON



(RPAQQ TEXTMENUICONMASK



;; TEdit-specific support

(DEFINEQ

(\TEDITMENU.CREATE

[LAMBDA NIL

(* gbn "27-Sep-84 01:04")
(* Creates the TEdit Expanded Menu)

(SETQ TEDIT.EXPANDED.MENU (\TEXTMENU.DOC.CREATE TEDIT.EXPANDEDMENU.SPEC])

(\TEDIT.EXPANDED.MENU

[LAMBDA (STREAM)

; Edited 20-Aug-87 16:51 by jds
(* "27-Sep-84 01:04" gbn)

(PROG (CHARMENUTEXTSTREAM)
(\TEXTMENU.START (SETQ CHARMENUTEXTSTREAM (COPYTEXTSTREAM TEDIT.EXPANDED.MENU T))
(\TEDIT.PRIMARYW (TEXTOBJ STREAM))
"Tedit Menu"
(HEIGHTIFWINDOW 60 T))

(COND
(OR (TEXTPROP STREAM 'CLEARGET)
(TEXTPROP STREAM 'CLEARPUT)) (* initialise the button)
(\TEXTMENU.SET.TOGGLE "Unformatted" 'ON CHARMENUTEXTSTREAM])

(MB.DEFAULTBUTTON.FN

[LAMBDA (OBJ SEL W)

; Edited 30-Mar-94 15:46 by jds
; MBFN for TEdit default menu item buttons.

(PROG* ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
(MAINTEXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
'TEXTOBJ))
(MAINSEL (fetch (TEXTOBJ SEL) of MAINTEXT))
OFILE CH PROC)
(COND
(EQ (fetch (TEXTOBJ EDITOPACTIVE) of MAINTEXT)
T)
(TEDIT.PROMPTPRINT MAINTEXT "Edit operation in progress; please wait." T)
(RETURN))
(fetch (TEXTOBJ EDITOPACTIVE) of MAINTEXT)
(TEDIT.PROMPTPRINT MAINTEXT (CONCAT (fetch (TEXTOBJ EDITOPACTIVE) of MAINTEXT)
" operation in progress; please wait.")
T)
[AND (NEQ (fetch (TEXTOBJ EDITOPACTIVE) of MAINTEXT)
(IMAGEOBJPROP OBJ 'MBTEXT])
(RETURN)))

[COND
(AND (SETQ PROC (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
'PROCESS))
(PROCESSP PROC)) ; THE MAIN window has a live process behind it; go evaluate
; the button fn there.

(PROCESS.EVAL PROC (LIST 'MB.DEFAULTBUTTON.ACTIONFN OBJ SEL W TEXTOBJ MAINTEXT MAINSEL)))
(AND (SETQ PROC (WINDOWPROP W 'PROCESS))
(PROCESSP PROC)) ; This window has a live process behind it; go evaluate the
; button fn there.

(PROCESS.EVAL PROC (LIST 'MB.DEFAULTBUTTON.ACTIONFN OBJ SEL W TEXTOBJ MAINTEXT MAINSEL)))
(T (ADD.PROCESS (LIST 'MB.DEFAULTBUTTON.ACTIONFN OBJ SEL W TEXTOBJ MAINTEXT MAINSEL])

(COND
(fetch (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ)
(GIVE.TTY.PROCESS W)
(DISMISS 20)))

[COND
(OR (fetch (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ)
(EQ (WINDOWPROP W 'PROCESS)
(TTY.PROCESS)))

; If the TEDIT MENU still has the tty, give it back to the real

; TEdit.

(SETQ TEDIT.SELPENDING NIL)
(GIVE.TTY.PROCESS (WINDOWPROP W 'MAINWINDOW)

:: Tell the menu button handler not to turn off this button--it's still active and will turn itself off.

(RETURN 'DON'T))

(TEDITMENU.RECORD.UNFORMATTED

(* jds " 7-Feb-85 09:44")

[LAMBDA (BUTTON NEWSTATE TEXTSTREAM)
(PROG ((FLG (COND
((EQ NEWSTATE 'ON)
T)
(T NIL)))
(TEXTOBJ (TEXTOBJ TEXTSTREAM))
(TEXTPROP TEXTOBJ 'UNFORMATTEDPUT/GET FLG))

(MB.DEFAULTBUTTON.ACTIONFN

; Edited 30-Mar-94 16:04 by jds
; MBFN for TEdit default menu item buttons.

[LAMBDA (OBJ SEL W TEXTOBJ MAINTEXT MAINSEL)
(PROG (OFILE CH %#COPIES PRINTHOST PRINTOPTIONS %#SIDES MSG)
[ERSETQ (RESETLST
[RESETSAVE (\TEDIT.MARKACTIVE MAINTEXT)
'(AND (\TEDIT.MARKINACTIVE OLDVALUE]
[RESETSAVE (\TEDIT.MARKACTIVE TEXTOBJ)
'(AND (\TEDIT.MARKINACTIVE OLDVALUE]
[RESETSAVE (PROG1 OBJ
(IMAGEOBJPROP OBJ 'MENUBUTTON.SELECTED T))
'(AND (IMAGEOBJPROP OLDVALUE 'MENUBUTTON.SELECTED NIL)
(replace (TEXTOBJ EDITOPACTIVE) of MAINTEXT with (OR (IMAGEOBJPROP OBJ 'MBTEXT)
T))
; So we can tell the guy WHAT op is active.
(SELECTQ (IMAGEOBJPROP OBJ 'MBTEXT)
(Put [SETQ OFILE (\TEDIT.MAKEFILENAME (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ
(fetch (SELECTION CH#) of SEL)
[COND
(OFILE ; Only try this if he really typed a file name
(TEDIT.PUT MAINTEXT OFILE NIL (TEXTPROP TEXTOBJ 'UNFORMATTEDPUT/GET])
(Get [SETQ OFILE (\TEDIT.MAKEFILENAME (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ
(fetch (SELECTION CH#) of SEL)
[COND
(OFILE ; Only try this if he really typed a file name
(TEDIT.GET MAINTEXT OFILE (TEXTPROP TEXTOBJ 'UNFORMATTEDPUT/GET])
(Include [SETQ OFILE (\TEDIT.MAKEFILENAME (MBUTTON.NEXT.FIELD.AS.TEXT
TEXTOBJ
(fetch (SELECTION CH#) of SEL)
(COND
(OFILE (TEDIT.INCLUDE MAINTEXT OFILE))))
(Find (SETQ OFILE (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ (fetch (SELECTION CH#)
of SEL)))
[COND
((ZEROP (NCHARS OFILE)) ; NOTHING--HE HIT DEL.
)
(OFILE ; There's something to do. Go do it.
(TEDIT.PROMPTPRINT MAINTEXT "Searching..." T)
[SETQ CH (CAR (ERSETQ (TEDIT.FIND MAINTEXT OFILE NIL NIL T)
(COND
(CH ; We found the target text.
(TEDIT.PROMPTPRINT MAINTEXT "Done.")
(\SHOWSEL MAINSEL NIL NIL)
(replace (SELECTION CH#) of MAINSEL with (CAR CH))
; Set up SELECTION to be the found text
(replace (SELECTION CHLIM) of MAINSEL with (ADD1 (CADR CH)))
(replace (SELECTION DCH) of MAINSEL
with (ADD1 (IDIFFERENCE (CADR CH)
(CAR CH)
(replace (SELECTION POINT) of MAINSEL with 'RIGHT)
(replace (TEXTOBJ CARETLOOKS) of MAINTEXT
with (\TEDIT.GET.INSERT.CHARLOOKS MAINTEXT MAINSEL))
; Set the caret looks to match those of the new selection
(TEDIT.RESET.EXTEND.PENDING.DELETE MAINSEL)
; And never pending a deletion.
(\FIXSEL MAINSEL MAINTEXT)
(TEDIT.NORMALIZECARET MAINTEXT MAINSEL)
(\SHOWSEL MAINSEL NIL T))
(T (TEDIT.PROMPTPRINT MAINTEXT "(Not found)"))
(Substitute [PROG* ((SAVECH# (fetch (SELECTION CH#) of SEL))
(REPLACEMENT (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ
(fetch (SELECTION CH#) of SEL))
[PATTERN (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ
(fetch (SELECTION CHLIM) of (fetch (TEXTOBJ
SCRATCHSEL
)
of TEXTOBJJ)
CONFIRM? KEEPLOOKS? LOC)

```

[SETQ LOC (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ
           (fetch (SELECTION CHLIM) of (fetch (TEXTOBJ
                                             SCRATCHSEL
                                             )
                                             of TEXTOBJ])
           )
[SETQ CONFIRM? (EQ 'ON (IMAGEOBJPROP (CAR LOC)
                                     'STATE)]
[SETQ LOC (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR LOC)
[SETQ KEEPLOOKS? (EQ 'ON (IMAGEOBJPROP (CAR LOC)
                                     'STATE)]
(COND
  ((ZEROP (NCHARS PATTERN))
   ;NOTHING--HE HIT DEL.
  )
  (PATTERN
   ; There's something to do. Go do it.
   [COND
    (KEEPLOOKS? (SETQ REPLACEMENT (
      MBUTTON.NEXT.FIELD.AS.PIECES
      TEXTOBJ SAVECH#])
    (RESETLST
     (RESETSAVE (CURSOR WAITINGCURSOR))
     (TEDIT.SUBSTITUTE (fetch (TEXTOBJ STREAMHINT)
                             of MAINTEXT)
                       PATTERN REPLACEMENT CONFIRM?)))
    ; He wants to QUIT the edit.
  (Quit
   (COND
    ((\TEDIT.QUIT (\TEDIT.PRIMARYW MAINTEXT)
     T)
     (replace (TEXTOBJ EDITFINISHEDFLG) of TEXTOBJ with T))))
  (Page% Layout
   ; Page layout menu
   (\TEXTMENU.START (COPYTEXTSTREAM TEDIT.EXPANDED.PAGEMENU T)
    (\TEDIT.PRIMARYW MAINTEXT)
    "Page Layout Menu"
    (HEIGHTIFWINDOW 135 5)))
  (Para% Looks
   ; Page layout menu
   (\TEDIT.EXPANDEDPARA.MENU MAINTEXT))
  (Char% Looks
   ; Page layout menu
   (\TEDIT.EXPANDEDCHARLOOKS.MENU MAINTEXT))
  (All
   ; Select the entire document.
   (COND
    ((NOT (ZEROP (fetch (TEXTOBJ TEXTLEN) of MAINTEXT)))
     (\SHOWSEL MAINSEL NIL NIL)
     (TEDIT.RESET.EXTEND.PENDING.DELETE MAINSEL)
     (replace (SELECTION CH#) of MAINSEL with 1)
     (replace (SELECTION CHLIM) of MAINSEL with (ADD1 (fetch (TEXTOBJ TEXTLEN)
                                                             of MAINTEXT)))
     (replace (SELECTION DCH) of MAINSEL with (fetch (TEXTOBJ TEXTLEN) of MAINTEXT))
     (replace (SELECTION POINT) of MAINSEL with 'LEFT)
     (replace (SELECTION SET) of MAINSEL with T)
     (\FIXSEL MAINSEL MAINTEXT)
     (\SHOWSEL MAINSEL NIL T))))
  (Hardcopy [SETQ PRINTHOST (\TEDIT.MAKEFILENAME (MBUTTON.NEXT.FIELD.AS.TEXT
      TEXTOBJ
      (fetch (SELECTION CH#) of SEL]
    (COND
     ((NOT PRINTHOST)
      ; If he didn't specify a particular host, defer to his defaults.
      (TEDIT.PROMPTPRINT MAINTEXT "Using default print server.)))
    [SETQ %#COPIES (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ
                  (fetch (SELECTION CH#) of (fetch (TEXTOBJ SCRATCHSEL)
                                                    of TEXTOBJ])
                  ; Grab the field that specifies number of copies.
    [COND
     (%#COPIES (SETQ PRINTOPTIONS (LIST '%#COPIES %#COPIES)
     (SETQ %#SIDES (SELECTQ (IMAGEOBJPROP
                          [CAR (MBUTTON.FIND.NEXT.BUTTON
                          TEXTOBJ
                          (fetch (SELECTION CHLIM)
                          of (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ])
                          'STATE)
                          (One% Side 1)
                          (Duplex 2)
                          NIL))
     [COND
     (%#SIDES (push PRINTOPTIONS %#SIDES)
              (push PRINTOPTIONS '%#SIDES)]
    [SETQ MSG (\TEDIT.MAKEFILENAME (MBUTTON.NEXT.FIELD.AS.TEXT
      TEXTOBJ
      (fetch (SELECTION CH#)
      of (fetch (TEXTOBJ SCRATCHSEL)
                of TEXTOBJ])
    [COND
     (MSG (push PRINTOPTIONS MSG)
          (push PRINTOPTIONS 'MESSAGE]
     (TEDIT.HARDCOPY MAINTEXT NIL NIL NIL PRINTHOST PRINTOPTIONS))
     (ERROR)))
  (replace (SELECTION SET) of SEL with T) ; Now turn the menu button highlighting off.

```

(replace (SELECTION ONFLG) of SEL with T)
(\SHOWSEL SEL NIL NIL)
(replace (SELECTION SET) of SEL with NIL) ; And forget that anything is selected.

])

)

(DEFINEQ

(\TEDIT.CHARLOOKSMENU.CREATE

[LAMBDA NIL

(SETQ TEDIT.CHARLOOKS.MENU (\TEXTMENU.DOC.CREATE (APPEND (LIST (create MB.BUTTON
MBLABEL _ 'APPLY
MBBUTTONEVENTFN _
'\TEDIT.APPLY.CHARLOOKS)
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ 'SHOW
MBBUTTONEVENTFN _
'\TEDIT.SHOW.CHARLOOKS)
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ 'NEUTRAL
MBBUTTONEVENTFN _
'\TEDIT.NEUTRALIZE.CHARLOOKS)
(create MB.TEXT
MBSTRING _ "
")
TEDIT.CHARLOOKSMENU.SPEC])

(\TEDIT.EXPANDEDCHARLOOKS.MENU

[LAMBDA (STREAM)

:: Open a character-looks menu.

(\TEXTMENU.START (COPYTEXTSTREAM TEDIT.CHARLOOKS.MENU T)
(\TEDIT.PRIMARYW STREAM)
"Character Looks Menu"
(HEIGHTIFWINDOW 68 T])

; Edited 20-Aug-87 16:49 by jds

(\TEDIT.APPLY.BOLDNESS

[LAMBDA (BUTTON NEWLOOKS)

(SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
(ON (CONS 'WEIGHT (CONS 'BOLD NEWLOOKS)))
(OFF (CONS 'WEIGHT (CONS 'MEDIUM NEWLOOKS)))
NEWLOOKS])

(* jds "30-Aug-84 13:55")

(\TEDIT.APPLY.CHARLOOKS

[LAMBDA (OBJ SEL W)

(PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
(MAINTTEXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
'TEXTOBJ))
(CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
SCRATCHSEL OFILE CH NEWLOOKS SIZE SUPER SUB NEXTB BUTTON TEXT OFFSET)
[SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
(* Skip over the SHOW button)
(* And over the NEUTRAL button.)
[SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
(* Now Parse the menu, to give us a looks spec.)
(TEDIT.LOOKS MAINTTEXT NEWLOOKS (fetch (SELECTION CH#) of (fetch (TEXTOBJ SEL) of MAINTTEXT))
(fetch (SELECTION DCH) of (fetch (TEXTOBJ SEL) of MAINTTEXT)))
(* Make the change in looks)
(* And turn off the APPLY button.)
(\SHOWSEL SEL NIL NIL)
(TTY.PROCESS (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
'PROCESS))
(* Leave him typing in the real document)

; Edited 30-May-91 22:17 by jds
(* MBFN for TEdit default menu item buttons.)

(\TEDIT.APPLY.OLINE

[LAMBDA (BUTTON NEWLOOKS)

(SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
(ON (CONS 'OVERLINE (CONS 'ON NEWLOOKS)))
(OFF (CONS 'OVERLINE (CONS 'OFF NEWLOOKS)))
NEWLOOKS])

(* jds "30-Aug-84 13:56")

(\TEDIT.SHOW.CHARLOOKS

[LAMBDA (OBJ SEL W)

:: Set the CHARLOOKS menu from the looks of the currently selected character.

; Edited 30-May-91 22:17 by jds

```
(LET* ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
      (MAINTXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
                          'TEXTOBJ))
      (MAINCH# (fetch (SELECTION CH#) of (fetch (TEXTOBJ SEL) of MAINTXT)))
      (CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
      (SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
      (PC OFILE CH NEWLOOKS NEXTB BUTTON TEXT OFFSET)
      (COND
        ((<= MAINCH# (fetch (TEXTOBJ TEXTLEN) of MAINTXT))
         [SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
                          ; Skip over the NEUTRAL button.

          (\SHOWSEL SEL NIL NIL)
          (replace (SELECTION SET) of SEL with NIL)
          (SETQ PC (\CHTOPC MAINCH# (fetch (TEXTOBJ PCTB) of MAINTXT)))
          ; The PIECE containing the text to describe
          ; Get the looks for those characters.
          (SETQ NEWLOOKS (fetch (PIECE PLOOKS) of PC))
          ; Get the looks for those characters.
          (WITHOUT-UPDATES TEXTOBJ SCRATCHSEL (\TEDIT.FILL.IN.CHARLOOKS.MENU TEXTOBJ CH# NEWLOOKS))
          ; Fill in the menu blanks with that info

        ]))
```

(\TEDIT.NEUTRALIZE.CHARLOOKS

[LAMBDA (OBJ SEL W) ; Edited 30-May-91 22:18 by jds

(* Handle the NEUTRAL button on a character looks menu. Sets all the menu settings neutral.)

```
(PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
      (MAINTXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
                          'TEXTOBJ))
      (CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
      (PC SCRATCHSEL OFILE CH NEWLOOKS NEXTB BUTTON TEXT OFFSET)
      (\SHOWSEL SEL NIL NIL)
      (replace (SELECTION SET) of SEL with NIL)
      (\TEDIT.NEUTRALIZE.CHARLOOKS.MENU TEXTOBJ CH#) (* Fill in the menu blanks with that info)
      (TEDIT.UPDATE.SCREEN TEXTOBJ) (* And update the screen image.)
      ])
```

(\TEDIT.FILL.IN.CHARLOOKS.MENU

[LAMBDA (TEXTOBJ CH# NEWLOOKS) ; Edited 30-May-91 22:28 by jds

;; Given a TEXTOBJ describing a charlooks menu, the CH# of the start of the charlooks menu, and a set of looks, fill in the menu fields.

```
(PROG (PC SCRATCHSEL OFILE CH NEXTB BUTTON TEXT OFFSET)
      (SETQ NEWLOOKS (\TEDIT.PARSE.CHARLOOKS.LIST NEWLOOKS NIL NIL))
      ;; Make sure the charlooks are in the proper internal format, so this fn can be called from every reasonable place.
      (SETQ SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
      [for PROP in (LIST (fetch (CHARLOOKS CLBOLD) of NEWLOOKS)
                       (fetch (CHARLOOKS CLITAL) of NEWLOOKS)
                       (fetch (CHARLOOKS CLULINE) of NEWLOOKS)
                       (fetch (CHARLOOKS CLSTRIKE) of NEWLOOKS)
                       (fetch (CHARLOOKS CLOLINE) of NEWLOOKS))
      do (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
      [COND
        (PROP ; Must set the property
          (IMAGEOBJPROP (CAR NEXTB)
                        'STATE
                        'ON)
          (T ; Must reset it.
            (IMAGEOBJPROP (CAR NEXTB)
                          'STATE
                          'OFF)
            (SETQ CH# (ADD1 (CDR NEXTB)
                          (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)) ; Get to the start of the text.
                          (SETQ BUTTON (CAR NEXTB))
                          [for ITEM in (IMAGEOBJPROP BUTTON 'BUTTONS)
                          do ; Loop thru the font FAMILY name button list, looking for one that
                          ; matches this text's looks
                          (COND
                            ((STRING-EQUAL [COND
                              ((AND (type? FONTCLASS (fetch (CHARLOOKS CLFONT) of NEWLOOKS))
                                   (NEQ (fetch FONTCLASSNAME of (fetch (CHARLOOKS CLFONT) of NEWLOOKS))
                                       'DEFAULTFONT))
                                   (CONCAT (fetch FONTCLASSNAME of (fetch (CHARLOOKS CLFONT) of NEWLOOKS))
                                       '-class))
                                   ((FONTP (fetch (CHARLOOKS CLFONT) of NEWLOOKS))
                                    (FONTPROP (fetch (CHARLOOKS CLFONT) of NEWLOOKS)
                                               'FAMILY]
                                   ITEM)
                                   (IMAGEOBJPROP BUTTON 'STATE ITEM)
                                   (RETURN)))
                            finally ; This font wasn't found in the list. Add it.
                              [MB.NWAYBUTTON.ADDITEM BUTTON (COND
                                ((type? FONTCLASS (fetch (CHARLOOKS CLFONT) of NEWLOOKS))
                                 (PACK* (fetch FONTCLASSNAME of (fetch (CHARLOOKS CLFONT)
```

```

; of NEWLOOKS))
' -class))
((FONTP (fetch (CHARLOOKS CLFONT) of NEWLOOKS))
(FONTPROP (fetch (CHARLOOKS CLFONT) of NEWLOOKS)
' FAMILY]
; Add this family to the list of items
(IMAGEOBJPROP BUTTON 'STATE (U-CASE (FONTPROP (fetch (CHARLOOKS CLFONT) of NEWLOOKS)
' FAMILY]
; Now find which text button was 'on'
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (CDR NEXTB))
NIL) ; Clean out the 'other font' field
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
(fetch (CHARLOOKS CLSIZE) of NEWLOOKS)) ; Set the value in the SIZE field
[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
; Move forward to the SUPERSCRIPT/SUBSCRIPT button
(SETQ BUTTON (CAR NEXTB))
(SETQ OFFSET (fetch (CHARLOOKS CLOFFSET) of NEWLOOKS)) ; Remember the offset value for later
[COND
((OR (NOT (fetch (CHARLOOKS CLOFFSET) of NEWLOOKS))
(ZEROP (fetch (CHARLOOKS CLOFFSET) of NEWLOOKS))))
; There is no subscript or superscript. Mark the text NORMAL.
(IMAGEOBJPROP BUTTON 'STATE 'Normal)
(SETQ OFFSET NIL) ; Mark there as being no offset value
)
((ILESSP OFFSET 0) ; SUBSCRIPTING
(IMAGEOBJPROP BUTTON 'STATE 'Subscript))
((IGREATERP OFFSET 0) ; SUBSCRIPTING
(IMAGEOBJPROP BUTTON 'STATE 'Superscript])
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (CDR NEXTB))
(AND OFFSET (IABS OFFSET))) ; Now move up to the offset distance fill-in field.
(\SHOWSEL SCRATCHSEL NIL NIL)
(replace (SELECTION SET) of SCRATCHSEL with NIL)
(\TEDIT.MARK.LINES.DIRTY TEXTOBJ CH# (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)))

```

(\TEDIT.NEUTRALIZE.CHARLOOKS.MENU

[LAMBDA (TEXTOBJ CH#) ; Edited 30-May-91 22:18 by jds

:: Set all the fields in the CHARLOOKS menu specified by TEXTOBJ, starting at CH# to neutral values.

```

(PROG ((SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
PC OFILE CH NEXTB BUTTON TEXT OFFSET)
(WITHOUT-UPDATES TEXTOBJ SCRATCHSEL [for PROP in '(BOLD ITAL ULINE STRIKE OLINE)
do (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
(IMAGEOBJPROP (CAR NEXTB)
' STATE
' NEUTRAL)
(SETQ CH# (ADD1 (CDR NEXTB)
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
; Get to the start of the text.
(SETQ BUTTON (CAR NEXTB))
(IMAGEOBJPROP BUTTON 'STATE NIL) ; Now find which text button was 'on'
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (CDR NEXTB))
NIL) ; Clean out the 'other font' field
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
NIL) ; Set the value in the SIZE field
[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
; Move forward to the SUPERSCRIPT/SUBSCRIPT button
(SETQ BUTTON (CAR NEXTB)) ; Remember the offset value for later
(IMAGEOBJPROP BUTTON 'STATE NIL)
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (CDR NEXTB))
NIL) ; Now move up to the offset distance fill-in field.
)
)
)

```

(\TEDIT.PARSE.CHARLOOKS.MENU

[LAMBDA (TEXTOBJ CH#) ; Edited 30-May-91 22:18 by jds
(* MBFN for TEdit default menu item buttons.)

```

(PROG (SCRATCHSEL CH NEWLOOKS SIZE SUPER SUB NEXTB BUTTON TEXT OFFSET)
(SETQ SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
[for BUTTON in '(BOLD ITALIC UNDERLINE STRIKEOUT OVERSCORE)
do (* Set the character properties which are independent)
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
(SELECTQ BUTTON
(BOLD (SETQ NEWLOOKS (\TEDIT.APPLY.BOLDNESS (CAR NEXTB)
NEWLOOKS)))
(ITALIC (SETQ NEWLOOKS (\TEDIT.APPLY.SLOPE (CAR NEXTB)
NEWLOOKS)))
(UNDERLINE (SETQ NEWLOOKS (\TEDIT.APPLY.ULINE (CAR NEXTB)
NEWLOOKS)))
(STRIKEOUT (SETQ NEWLOOKS (\TEDIT.APPLY.STRIKEOUT (CAR NEXTB)
NEWLOOKS)))
(OVERSCORE (SETQ NEWLOOKS (\TEDIT.APPLY.OLINE (CAR NEXTB)
NEWLOOKS)))
NIL)
(SETQ CH# (ADD1 (CDR NEXTB)

```

```

(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)) (* Get to the start of the text.)
(SETQ BUTTON (CAR NEXTB))
[AND BUTTON (SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
  (Other (* Have to get and add in a new font.)
    (COND
      ([SETQ TEXT (MBUTTON.NEXT.FIELD.AS.ATOM TEXTOBJ (ADD1 (CDR NEXTB)
        (* He wants some font not on the list.
        Add it to the list.)
        (SETQ NEWLOOKS (CONS 'FAMILY (CONS (U-CASE TEXT)
          NEWLOOKS)))
        (COND
          ([NOT (FMEMB (U-CASE TEXT)
            (U-CASE (IMAGEOBJPROP BUTTON 'BUTTONS)
              (* This font name isn't in the list already;
              add it.)
              (MB.NWAYBUTTON.ADDITEM BUTTON TEXT)
              (IMAGEOBJPROP BUTTON 'STATE TEXT))
            (T [IMAGEOBJPROP BUTTON 'STATE (for NAME
              in (IMAGEOBJPROP BUTTON 'BUTTONS)
              suchthat (EQ (U-CASE TEXT)
                (U-CASE NAME)
              (* Select the newly-specified font.)
            ))
          (TEDIT.DELETE TEXTOBJ SCRATCHSEL)
            (* Delete the new font's name from the fill-in field.)
          (TEDIT.OBJECT.CHANGED TEXTOBJ BUTTON))
            (* He didn't specify a font. Complain but keep on.)
          (TEDIT.PROMPTPRINT TEXTOBJ "'Other' font not specified; no change
            made." T))))
        (COND
          ((STRPOS '-class (IMAGEOBJPROP BUTTON 'STATE))
            (* It's a font class. Grab the name and evaluate it.)
          (SETQ NEWLOOKS
            (CONS 'FONT (CONS [EVAL (MKATOM (SUBSTRING (IMAGEOBJPROP BUTTON 'STATE)
              1
              (SUB1 (STRPOS '-class (IMAGEOBJPROP
                BUTTON
                'STATE]
                NEWLOOKS)))
            (MBUTTON.FIND.NEXT.FIELD TEXTOBJ CH#))
          (T (SETQ NEWLOOKS (CONS 'FAMILY (CONS (U-CASE (IMAGEOBJPROP BUTTON 'STATE)
            NEWLOOKS)))
            (MBUTTON.FIND.NEXT.FIELD TEXTOBJ CH#)
            (* Skip over the "other text" fill-in.)
            (* Now find which text button was "on")
            (fetch (SELECTION CH#) of SCRATCHSEL]
            (* Read the contents of the SIZE menu field)
          ]
          (SETQ SIZE (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1
            (* He specified one. Set it.)
            (SIZE
              (SETQ NEWLOOKS (CONS 'SIZE (CONS SIZE NEWLOOKS)
            (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
            (* Get a handle on the SUPERSCRIPT/SUBSCRIPT button)
            (SETQ BUTTON (CAR NEXTB))
            (SETQ SUPER (IMAGEOBJPROP BUTTON 'STATE))
            (* Decide which kind it is)
            (SETQ OFFSET (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (CDR NEXTB)
            (* And get the offset distance, in points.)
            (SELECTQ SUPER
              (Superscript
            (* He called for SUPERSCRIPTing. Offset the characters by either the distance he gave, or 2 pts.)
            (SETQ NEWLOOKS (CONS 'SUPERSCRIPT (CONS (OR OFFSET 2)
              NEWLOOKS)))
            (Subscript
            (* He called for SUBSCRIPTING. Offset the characters by either the distance he gave, or 2 pts if he gave no distance.)
            (SETQ NEWLOOKS (CONS 'SUBSCRIPT (CONS (OR OFFSET 2)
              NEWLOOKS)))
            (Normal
              (* NORMAL => Turn off all super and subscripting)
            (SETQ NEWLOOKS (CONS 'SUPERSCRIPT (CONS 0 NEWLOOKS)))
            NIL)
            (RETURN NEWLOOKS])

```

(\TEDIT.APPLY.SLOPE

(* jds "30-Aug-84 13:56")

```

[LAMBDA (BUTTON NEWLOOKS)
  (SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
    (ON (CONS 'SLOPE (CONS 'ITALIC NEWLOOKS)))
    (OFF (CONS 'SLOPE (CONS 'REGULAR NEWLOOKS)))
    NEWLOOKS])

```

(\TEDIT.APPLY.STRIKEOUT

(* jds "30-Aug-84 13:56")

```

[LAMBDA (BUTTON NEWLOOKS)
  (SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
    (ON (CONS 'STRIKEOUT (CONS 'ON NEWLOOKS)))

```


(OFF (CONS 'STRIKEOUT (CONS 'OFF NEWLOOKS)))
NEWLOOKS])

(\TEDIT.APPLY.ULINE

[LAMBDA (BUTTON NEWLOOKS) ;* jds "30-Aug-84 13:56"
(SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
(ON (CONS 'UNDERLINE (CONS 'ON NEWLOOKS)))
(OFF (CONS 'UNDERLINE (CONS 'OFF NEWLOOKS)))
NEWLOOKS])

(DEFINEQ

(\TEDITPARAMENU.CREATE

[LAMBDA NIL ;* jds " 2-Aug-84 15:32"
;* Creates the TEdit Expanded Paragraph Menu)
(SETQ TEDIT.EXPANDEDPARA.MENU (\TEXTMENU.DOC.CREATE TEDIT.PARAMENU.SPEC])

(\TEDIT.EXPANDEDPARA.MENU

[LAMBDA (STREAM) ; Edited 20-Aug-87 16:51 by jds
(\TEXTMENU.START (COPYTEXTSTREAM TEDIT.EXPANDEDPARA.MENU T)
(\TEDIT.PRIMARYW (TEXTOBJ STREAM))
"Paragraph-Looks Menu"
(HEIGHTIFWINDOW 141 T])

(\TEDIT.APPLY.PARALOOKS

[LAMBDA (OBJ SEL W) ; Edited 22-Apr-93 16:45 by jds

:: Handler for the Paragraph Menu's APPLY button. Collects the specs from the paragraph menu and calls TEDIT.PARALOOKS to effect the
:: change.

(PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
(MAINTXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
'TEXTOBJ))
(CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
SCRATCHSEL QUAD OFILE CH NEWLOOKS SIZE SUPER SUB LINELEAD PARALEAD DEFAULTTAB BUTTON NEXTB BUTTONDATA
L1 LN R PARATYPE SPECIALX SPECIALY)
[SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
; Skip the SHOW button
[SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
; and the NEUTRAL button.
(SETQ SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
(SETQ NEWLOOKS NIL) ; The list we'll be collecting the looks changes in.
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)) ; Get the JUSTIFICATION button: Left/Right/Centered/Justified
(SETQ BUTTON (CAR NEXTB))
[COND
((AND (SETQ QUAD (IMAGEOBJPROP BUTTON 'STATE))
(NEQ QUAD 'OFF)) ; A justification was specified
(SETQ NEWLOOKS (CONS 'QUAD (CONS (U-CASE (MKATOM QUAD))
NEWLOOKS])
[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB])
; Go to the 'Page Heading' button
(SETQ BUTTON (CAR NEXTB))
[COND
((EQ (IMAGEOBJPROP BUTTON 'STATE)
'ON) ; This paragraph IS a page heading.
(SETQ NEWLOOKS (CONS 'TYPE (CONS 'PAGEHEADING NEWLOOKS)))
; Tell him that it's a heading.
(SETQ NEWLOOKS (CONS 'SUBTYPE (CONS [MKATOM (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ
(ADD1 (CDR NEXTB])
NEWLOOKS))) ; And say what kind.
)
((EQ (IMAGEOBJPROP BUTTON 'STATE)
'OFF) ; This paragraph IS NOT a page heading.
(SETQ NEWLOOKS (CONS 'TYPE (CONS NIL NEWLOOKS))) ; Tell him that it's NOT a heading.
(SETQ NEWLOOKS (CONS 'SUBTYPE (CONS NIL NEWLOOKS)))
(MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ (ADD1 (CDR NEXTB)))
; And say what kind.
)
(T ; No change specified. Skip over the heading-type fill-in.
(MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ (ADD1 (CDR NEXTB])
[COND
((SETQ LINELEAD (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (fetch (SELECTION CH#) of SCRATCHSEL)))
; Get any line leading
(SETQ NEWLOOKS (CONS 'LINELEADING (CONS LINELEAD NEWLOOKS])
[COND
([SETQ PARALEAD (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL])
; Get any paragraph leading
(SETQ NEWLOOKS (CONS 'PARALEADING (CONS PARALEAD NEWLOOKS])
[COND
([SETQ SPECIALX (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL])
; Get any special X position for the paragraph

```

      (SETQ NEWLOOKS (CONS 'SPECIALX (CONS (FIXR (TIMES 12 SPECIALX))
                                          NEWLOOKS])
[COND
  ([SETQ SPECIALY (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
                                                                ; Get special Y positioning for the paragraph
                                                                (SETQ NEWLOOKS (CONS 'SPECIALY (CONS (FIXR (TIMES 12 SPECIALY))
                                          NEWLOOKS])
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (fetch (SELECTION CH#) of SCRATCHSEL)))
(SETQ BUTTON (CAR NEXTB))
[COND
  [(EQ (IMAGEOBJPROP BUTTON 'STATE)
       'ON)
       ; This paragraph starts on a new page (or col or box, as
       ; appropriate)
  (SETQ NEWLOOKS (CONS 'NEWPAGEBEFORE (CONS T NEWLOOKS])
  (EQ (IMAGEOBJPROP BUTTON 'STATE)
      'OFF)
      ; This paragraph IS NOT a page heading.
  (SETQ NEWLOOKS (CONS 'NEWPAGEBEFORE (CONS NIL NEWLOOKS])
[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB))
(SETQ BUTTON (CAR NEXTB))
[COND
  [(EQ (IMAGEOBJPROP BUTTON 'STATE)
       'ON)
       ; The next paragraph starts on a new page....
  (SETQ NEWLOOKS (CONS 'NEWPAGEAFTER (CONS T NEWLOOKS])
  (EQ (IMAGEOBJPROP BUTTON 'STATE)
      'OFF)
      ; The next paragraph DOESN'T START on a new page....
  (SETQ NEWLOOKS (CONS 'NEWPAGEAFTER (CONS NIL NEWLOOKS])
[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB))
(SETQ BUTTON (CAR NEXTB))
(SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
  (ON (push NEWLOOKS T)
      (push NEWLOOKS 'HARDCOPY))
  (OFF (push NEWLOOKS NIL)
       (push NEWLOOKS 'HARDCOPY))
NIL)

```

;;; THE VARIOUS KINDS OF KEEP PROPERTIES (ONLY HEADING-KEEP FOR NOW THO)

```

[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB))
(SETQ BUTTON (CAR NEXTB))
(SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
  (ON (push NEWLOOKS 'ON)
      (push NEWLOOKS 'HEADINGKEEP))
  (OFF (push NEWLOOKS 'OFF)
       (push NEWLOOKS 'HEADINGKEEP))
NIL)

```

;;; THE DEFAULT TAB WIDTH

```

(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (fetch (SELECTION CH#) of SCRATCHSEL)))
(SETQ BUTTON (CAR NEXTB))
(SETQ DEFAULTTAB (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (CDR NEXTB)))
(while (NOT (type? MARGINBAR BUTTON)) do (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ
                                                         (fetch (SELECTION CH#) of SCRATCHSEL)))
      (SETQ BUTTON (CAR NEXTB)))
(SETQ BUTTONDATA (IMAGEOBJPROP BUTTON 'OBJECTDATUM))
[COND
  ((IGEQL [SETQ L1 (FIXR (TIMES (fetch MARL1 of BUTTONDATA)
                          (fetch MARUNIT of BUTTONDATA)
                          0)
          ; The 1stleftmargin is set, and non-neutral.
  (SETQ NEWLOOKS (CONS '1STLEFTMARGIN (CONS L1 NEWLOOKS])
[COND
  ((IGEQL [SETQ LN (FIXR (TIMES (fetch MARLN of BUTTONDATA)
                          (fetch MARUNIT of BUTTONDATA)
                          0)
          ; The LEFTMARGIN is set, and non-neutral.
  (SETQ NEWLOOKS (CONS 'LEFTMARGIN (CONS LN NEWLOOKS])
[COND
  ((IGEQL [SETQ R (FIXR (TIMES (fetch MARR of BUTTONDATA)
                          (fetch MARUNIT of BUTTONDATA)
                          0)
          ; The RIGHTMARGIN is set, and non-neutral.
  (SETQ NEWLOOKS (CONS 'RIGHTMARGIN (CONS R NEWLOOKS])
[COND
  ((NEQL (fetch MARTABS of BUTTONDATA)
         'NEUTRAL)
         ; If the tab settings are neutral, don't change anything.
  (SETQ NEWLOOKS (CONS 'TABS
                      (CONS [CONS DEFAULTTAB
                          (SORT (for TAB in (fetch MARTABS of BUTTONDATA)
                                          collect (CONS (FIXR (TIMES (CAR TAB)
                                          (fetch MARUNIT of BUTTONDATA)))
                                          (CDR TAB))))
                          (FUNCTION (LAMBDA (A B)
                                      (ILEQL (CAR A)
                                              (CAR B)
                                              NEWLOOKS])
(TEDIT.PARALOOKS MAINTXT NEWLOOKS (fetch (SELECTION CH#) of (fetch (TEXTOBJ SEL) of MAINTXT))
  (fetch (SELECTION DCH) of (fetch (TEXTOBJ SEL) of MAINTXT)))
(\SHOWSEL SEL NIL NIL)
(TTY.PROCESS (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)

```

'PROCESS])

(\TEDIT.SHOW.PARALOOKS

[LAMBDA (OBJ SEL W)

; Edited 6-Jul-92 09:42 by jds

:: Fill in the PARAGRAPH LOOKS menu from the para looks for a selected character

```

(PROG* ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
  (MAINTEXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
    'TEXTOBJ))
  (CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
  (SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
  (OLDUPDATEFLG (fetch (TEXTOBJ TXTDON'TUPDATE) of TEXTOBJ))
  FMTSPEC BUTTON NEXTB ARB BUTTONDATA)
  (\SHOWSEL SEL NIL NIL)
  (replace (SELECTION SET) of SEL with NIL)
  (COND
    ((ZEROP (fetch (TEXTOBJ TEXTLEN) of MAINTEXT)) ; If there is no text to take the formatting from, don't bother
      (RETURN)))
  (WITHOUT-UPDATES TEXTOBJ SCRATCHSEL [SETQ FMTSPEC
    (fetch (PIECE PPARALOOKS)
      of (\CHTOPC [IMAX 1 (IMIN (fetch (TEXTOBJ TEXTLEN)
        of MAINTEXT)
        (fetch (SELECTION CH#)
          of (fetch (TEXTOBJ SEL)
            of MAINTEXT))
        (fetch (TEXTOBJ PCTB) of MAINTEXT])
        ; Get to the start of the text.
        [SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
          ; Skip the NEUTRAL button.
          (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
          ; Grab the justification button
          (SETQ BUTTON (CAR NEXTB))
          [for ITEM in (IMAGEOBJPROP BUTTON 'BUTTONS) do (COND
            ([EQ (fetch (FMTSPEC QUAD) of FMTSPEC)
              (U-CASE (COND
                ((LISTP ITEM)
                  (CAR ITEM))
                (T ITEM)
              )
              ; Turn this button on.
              (IMAGEOBJPROP BUTTON 'STATE ITEM)
              (RETURN)
              ; Now find which text button was 'on'
              [SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB)
                ; Find the 'Page Heading' button
                (SETQ BUTTON (CAR NEXTB))
                (COND
                  ((EQ (fetch (FMTSPEC FMTPARATYPE) of FMTSPEC)
                    'PAGEHEADING)
                    ; This IS a page heading. Turn the button ON and set the
                    ; heading type field
                    (IMAGEOBJPROP BUTTON 'STATE 'ON)
                    (MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
                      (fetch (FMTSPEC FMTPARASUBTYPE) of FMTSPEC)))
                    ; This isn't a page heading; make sure the type field is empty.
                    (T
                      (IMAGEOBJPROP BUTTON 'STATE 'OFF)
                      (MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
                        NIL)))
                    (MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
                      (fetch (FMTSPEC LINELEAD) of FMTSPEC)) ; Update the LINE LEADING field
                    (MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
                      (fetch (FMTSPEC LEADBEFORE) of FMTSPEC)) ; Update the PARA LEADING field
                    [MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
                      (LET ((VAL (/ (FIXR (IQUOTIENT (OR (fetch (FMTSPEC FMTSPECIALX) of FMTSPEC)
                        0)
                        3))
                        4)))
                        (COND
                          ((FIXP VAL)
                            VAL)
                          (T (FLOAT VAL)
                            [MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
                              (LET ((VAL (/ (FIXR (IQUOTIENT (OR (fetch (FMTSPEC FMTSPECIALY) of FMTSPEC)
                                0)
                                3))
                                4)))
                                  (COND
                                    ((FIXP VAL)
                                      VAL)
                                    (T (FLOAT VAL)
                                      [SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
                                        (SETQ BUTTON (CAR NEXTB))
                                        (COND
                                          ((fetch (FMTSPEC FMTNEWPAGEBEFORE) of FMTSPEC)
                                            (IMAGEOBJPROP BUTTON 'STATE 'ON)
                                            ; This para starts on a new page
                                          )
                                          (T (IMAGEOBJPROP BUTTON 'STATE 'OFF])

```

```

[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB)
(SETQ BUTTON (CAR NEXTB))
[COND
  ((fetch (FMTSPEC FMNEWPAGEAFTER) of FMTSPEC)
   (IMAGEOBJPROP BUTTON 'STATE 'ON) ; This para starts on a new page
  )
  (T (IMAGEOBJPROP BUTTON 'STATE 'OFF])
;; HARDCOPY-DISPLAY MODE
[SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ (ADD1 (CDR NEXTB))
(COND
  ((fetch (FMTSPEC FMTHARDCOPY) of FMTSPEC)
   'ON) ; This para is to be formatted for hardcopy on the display
  (T 'OFF])
;; HEADING KEEP
[SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ NEXTB (COND
  ((fetch (FMTSPEC FMTHEADINGKEEP)
   of FMTSPEC)
   'ON) ; This para is to be formatted for hardcopy on the display
  (T 'OFF])
;; DEFAULT TAB WIDTH
(MBUTTON.SET.NEXT.FIELD TEXTOBJ NEXTB (CAR (fetch (FMTSPEC TABSPEC) of FMTSPEC)))
; Update the DEFAULT TAB SPACING field
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (fetch (SELECTION CH#) of SCRATCHSEL)))
(SETQ BUTTON (CAR NEXTB))
(while (NOT (type? MARGINBAR BUTTON)) do [SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON
TEXTOBJ
(ADD1 (CDR NEXTB)
(SETQ BUTTON (CAR NEXTB)))
(SETQ BUTTONDATA (IMAGEOBJPROP BUTTON 'OBJECTDATUM))
; (IMAGEOBJPROP BUTTON (QUOTE IMAGECACHE) NIL)
; Tell it to reformat itself.
(replace MARL1 of BUTTONDATA with (FQUOTIENT (fetch (FMTSPEC 1STLEFTMAR) of FMTSPEC)
(fetch MARUNIT of BUTTONDATA)))
(replace MARLN of BUTTONDATA with (FQUOTIENT (fetch (FMTSPEC LEFTMAR) of FMTSPEC)
(fetch MARUNIT of BUTTONDATA)))
(replace MARR of BUTTONDATA with (FQUOTIENT (fetch (FMTSPEC RIGHTMAR) of FMTSPEC)
(fetch MARUNIT of BUTTONDATA)))
(replace MARTABS of BUTTONDATA with (for TAB in (CDR (fetch (FMTSPEC TABSPEC) of FMTSPEC))
collect (CONS (FQUOTIENT (CAR TAB)
(fetch MARUNIT of BUTTONDATA))
(CDR TAB]))

```

(\TEDIT.NEUTRALIZE.PARALOOKS.MENU

[LAMBDA (OBJ SEL W)

; Edited 30-May-91 22:18 by jds

;; Set all the fields of a PARAGRAPH LOOKS menu to neutral settings.

```

(PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
(CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
SCRATCHSEL FMTSPEC BUTTON NEXTB ARB BUTTONDATA)
(SETQ SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
(WITHOUT-UPDATES TEXTOBJ SCRATCHSEL (SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ CH#
'NIL))
; Neutralize the justification N-Way button
(SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ NEXTB 'NEUTRAL))
; Find the 'Page Heading' button
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
NIL)
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
NIL) ; Update the LINE LEADING field
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
NIL) ; Update the PARA LEADING field
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
NIL)
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
NIL)
(SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
'NEUTRAL)) ; New page before
(SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ NEXTB 'NEUTRAL)) ; New page after
(SETQ NEXTB (MBUTTON.SET.NEXT.BUTTON.STATE TEXTOBJ NEXTB 'NEUTRAL)) ; Hardcopy formatting mode
(MBUTTON.SET.NEXT.FIELD TEXTOBJ NEXTB NIL) ; Update the DEFAULT TAB SPACING field
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (fetch (SELECTION CH#) of SCRATCHSEL)))
(SETQ BUTTON (CAR NEXTB))
(while (NOT (type? MARGINBAR BUTTON)) do [SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON
TEXTOBJ
(ADD1 (CDR NEXTB)
(SETQ BUTTON (CAR NEXTB)))
(SETQ BUTTONDATA (IMAGEOBJPROP BUTTON 'OBJECTDATUM))

```

```

; (IMAGEOBJPROP BUTTON (QUOTE IMAGECACHE) NIL)
; Tell it to reformat itself.
[replace MARL1 of BUTTONDATA with (COND
  ((ILESSP (fetch MARL1 of BUTTONDATA)
    0)
    (fetch MARL1 of BUTTONDATA))
  (T (IMIN -0.5 (IMINUS (fetch MARL1 of BUTTONDATA)
    (ILESSP (fetch MARLN of BUTTONDATA)
      0)
    (fetch MARLN of BUTTONDATA))
    (T (IMIN -0.5 (IMINUS (fetch MARLN of BUTTONDATA)
      (ILESSP (fetch MARR of BUTTONDATA)
        0)
      (fetch MARR of BUTTONDATA))
      ((ZEROP (fetch MARR of BUTTONDATA)
        (IMINUS (IQUOTIENT (IDIFFERENCE (fetch (TEXTOBJ WRIGHT)
          of TEXTOBJ)
            20)
            12))))
      (T (IMIN -0.5 (IMINUS (fetch MARR of BUTTONDATA)
        (replace MARTABS of BUTTONDATA with 'NEUTRAL]))

```

(\TEDIT.RECORD.TABLEADERS

[LAMBDA (BUTTON NEWSTATE TEXTSTREAM SEL)

; Edited 30-May-91 22:18 by jds

(* Toggle the dotted-leader state of the margin bar tab-setter. This is called when the user hits the "dotted leader" toggle button in the menu)

```

(PROG* [(FLG (COND
  (EQ NEWSTATE 'ON)
  T)
  (T NIL)))
  (TEXTOBJ (TEXTOBJ TEXTSTREAM))
  (MARGINBAR (CAR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SEL)
    (replace MARTABTYPE of (IMAGEOBJPROP MARGINBAR 'OBJECTDATUM)
      with (SELECTQ (OR (fetch MARTABTYPE of (IMAGEOBJPROP MARGINBAR 'OBJECTDATUM))
        'LEFT)
        (LEFT 'DOTTEDLEFT)
        (DOTTEDLEFT 'LEFT)
        (CENTERED 'DOTTEDCENTERED)
        (DOTTEDCENTERED
          'CENTERED)
        (RIGHT 'DOTTEDRIGHT)
        (DOTTEDRIGHT 'RIGHT)
        (DECIMAL 'DOTTEDDECIMAL)
        (DOTTEDDECIMAL
          'DECIMAL)
        (SHOULDNT]))

```

(DEFINEQ

(\TEDIT.SHOW.PAGEFORMATTING

[LAMBDA (OBJ SEL W)

; Edited 4-Feb-92 16:38 by jds

:: Take a document's page formatting, and display it in the menu.

```

(PROG* ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
  (MAINTXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
    'TEXTOBJ))
  (CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
  (SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
  (OLDUPDATEFLG (fetch (TEXTOBJ TXTDON'TUPDATE) of TEXTOBJ))
  FOLIOINFO NEWLOOKS NEXTB BUTTON PAGEID OPAGEFRAMES FIRST REST PFONT HEADING HEADINGS PAGEPROPS
  STARTINGPAGE# PAPERSIZE)

```

:: Start by turning off the selection--and leaving it off afterward.

```

(\SHOWSEL SEL NIL NIL)
(replace (SELECTION SET) of SEL with NIL)

```

:: What kind of page are we looking at the specs for?

```

(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
(SETQ BUTTON (CAR NEXTB))
(SELECTQ (IMAGEOBJPROP (CAR NEXTB)
  'STATE)
  (|First(&Default)|
    (SETQ PAGEID 'FIRST))
  (Other% Left (SETQ PAGEID 'LEFT))
  (Other% Right (SETQ PAGEID 'RIGHT))
  (PROGN (TEDIT.PROMPTPRINT MAINTXT "First specify which kind of page you want to see." T)
    (SETQ PAGEID NIL)))

```

:: Now mark the menu for NO SCREEN UPDATES during the button-setting process.

(AND PAGEID (WITHOUT-UPDATES TEXTOBJ SCRATCHSEL (replace (TEXTOBJ TXTDON'TUPDATE) of TEXTOBJ with T)

;; Now replace the button values, fill-in fields, etc.

(SETQ OPAGEFRAMES (OR (fetch (TEXTOBJ TXTPAGEFRAMES) of MAINTTEXT) TEDIT.PAGE.FRAMES))

[COND ((LISTP OPAGEFRAMES) ; No problem, this is already just a list of first-recto-verso frames

) (T ; This is probably a parsed-up version of the thing. Fix it to a list.

(COND [(EQ (fetch (PAGEREGION REGIONFILLMETHOD) of OPAGEFRAMES) 'SEQUENCE)

(SETQ FIRST (CAR (fetch (PAGEREGION REGIONSUBBOXES) of OPAGEFRAMES))) (SETQ REST (CADR (fetch (PAGEREGION REGIONSUBBOXES) of OPAGEFRAMES)))

(COND [(EQ (fetch (PAGEREGION REGIONFILLMETHOD) of REST) 'ALTERNATE)

(SETQ OPAGEFRAMES (CONS FIRST (fetch (PAGEREGION REGIONSUBBOXES) of REST)

(T (SETQ OPAGEFRAMES NIL]

(T (SETQ OPAGEFRAMES NIL]

(COND ((NOT OPAGEFRAMES) ; If the formatting isn't in our simplified 3-way format, punt out of this.

(TEDIT.PROMPTPRINT MAINTTEXT "Format too complex to edit." T) (SETQ PAGEID NIL)))

(SELECTQ PAGEID (FIRST (SETQ NEWLOOKS (CAR OPAGEFRAMES))) (LEFT (SETQ NEWLOOKS (CADR OPAGEFRAMES)) (SETQ PAPERSIZE (LISTGET [CAR (FLAST (TEDIT.UNPARSE.PAGEFORMAT (CAR OPAGEFRAMES) 'PICAS]

'PAPERSIZE)))

(RIGHT (SETQ NEWLOOKS (CADDR OPAGEFRAMES)) (SETQ PAPERSIZE (LISTGET [CAR (FLAST (TEDIT.UNPARSE.PAGEFORMAT (CAR OPAGEFRAMES) 'PICAS]

'PAPERSIZE)))

NIL)

(COND (PAGEID (SETQ NEWLOOKS (TEDIT.UNPARSE.PAGEFORMAT NEWLOOKS 'PICAS)) (SETQ PAGEPROPS (CAR (FLAST NEWLOOKS)))

[COND ((EQ PAGEID 'FIRST) (SETQ PAPERSIZE (LISTGET PAGEPROPS 'PAPERSIZE] (SETQ CH# (ADD1 (CDR NEXTB)))

; Move past the kind-of-page button (SETQ STARTINGPAGE# (LISTGET PAGEPROPS 'STARTINGPAGE#)) ; Grab a potential starting page number.

(MBUTTON.SET.NEXT.FIELD TEXTOBJ CH# STARTINGPAGE#) (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)) (SETQ CH# (ADD1 (CDR NEXTB)))

(IMAGEOBJPROP (CAR NEXTB) 'STATE (OR PAPERSIZE 'Letter))

(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))

(SETQ CH# (ADD1 (CDR NEXTB)))

[IMAGEOBJPROP (CAR NEXTB)

'STATE (COND ((LISTGET PAGEPROPS 'LANDSCAPE?) 'ON)

(T 'OFF] ; Tell whether the page is to be landscape or not. (SETQ FOLIOINFO (LISTGET PAGEPROPS 'FOLIOINFO))

; Page number formatting info (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))

(SETQ CH# (ADD1 (CDR NEXTB)))

[IMAGEOBJPROP (CAR NEXTB)

'STATE (COND ((pop NEWLOOKS) 'Yes)

(T 'No] (SETQ BUTTON (CAR NEXTB)) (MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (CDR NEXTB)) (pop NEWLOOKS)) ; Page # X location

(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL

))

(pop NEWLOOKS)) ; Page # Y location

(SETQ PFONT (pop NEWLOOKS)) ; Skip the font info for now.

[SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL

(SETQ CH# (ADD1 (CDR NEXTB)))

```

(SETQ BUTTON (CAR NEXTB))
(IMAGEOBJPROP BUTTON 'STATE (SELECTQ (pop FOLIOINFO)
                                     (ARABIC 123)
                                     (LOWERROMAN 'xiv)
                                     (UPPERROMAN 'XIV)
                                     123))
                                     ; The format for the page number
(SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
(SETQ CH# (ADD1 (CDR NEXTB)))
(SETQ BUTTON (CAR NEXTB))
                                     ; How to align the page number
(IMAGEOBJPROP BUTTON 'STATE (SELECTQ (pop NEWLOOKS)
                                     (LEFT 'Left)
                                     (RIGHT 'Right)
                                     (CENTERED 'Centered)
                                     'Centered))
(MBUTTON.SET.NEXT.FIELD TEXTOBJ CH# (pop FOLIOINFO))
                                     ; The text to surround the page number
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop FOLIOINFO))
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; Left Margin
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; Right Margin
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; Top margin
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; Bottom Margin
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; # of columns
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; Column width
(MBUTTON.SET.NEXT.FIELD TEXTOBJ (ADD1 (fetch (SELECTION CH#) of
                                             SCRATCHSEL
                                             ))
                          (pop NEWLOOKS)) ; Intercolumn spacing
(SETQ HEADINGS (pop NEWLOOKS))
(for HEADING# from 1 to 8 do ;; Insert info about up to 8 headings (the # of spots in the
                           ;; menu)
  (SETQ HEADING (pop HEADINGS))
  (MBUTTON.SET.NEXT.FIELD
   TEXTOBJ
   (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
   (pop HEADING))
  (MBUTTON.SET.NEXT.FIELD
   TEXTOBJ
   (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
   (pop HEADING))
  (MBUTTON.SET.NEXT.FIELD
   TEXTOBJ
   (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL))
   (pop HEADING)))
(COND
 (HEADINGS
  ;; There were headings left over, so warn user.
  (PROMPTPRINT "WARNING: This document has more kinds of page
                heading than the menu has room for. Some will be lost
                if you APPLY this menu."))
 (\TEDIT.FILL.IN.CHARLOOKS.MENU TEXTOBJ (ADD1 (fetch (SELECTION CH#)
                                                    of SCRATCHSEL))
  (OR PFONT TEDIT.DEFAULT.FOLIO.LOOKS))
  ; The font for the page numbers to appear in.
)

```

(\TEDITPAGEMENU.CREATE

[LAMBDA NIL

(* gbn " 8-Oct-84 18:25")

(* Creates the TEdit Expanded Menu)

(SETQ TEDIT.EXPANDED.PAGEMENU (\TEXTMENU.DOC.CREATE (APPEND TEDIT.PAGEMENU.SPEC TEDIT.MENUDIVIDER.SPEC
[LIST (create MB.TEXT

MBSTRING _ "Character Looks for
Page Numbers: "
MBFONT _ (FONTCREATE 'HELVETICA 10
'BOLD]

TEDIT.CHARLOOKSMENU.SPEC])

(\TEDIT.APPLY.PAGEFORMATTING

[LAMBDA (OBJ SEL W)

; Edited 4-Jun-93 12:04 by sybalsky:mv:envos

::: Change the page formatting for this document

```
(PROG ((TEXTOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
  (MAINTTEXT (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
    'TEXTOBJ))
  (CH# (ADD1 (fetch (SELECTION CH#) of SEL)))
  SCRATCHSEL NEXTB BUTTON OPAGEFRAMES PAGEID PX PY LEFT BOTTOM TOP RIGHT ALIGNMENT PAGENOS COLS
  COLWIDTH INTERCOL PFONT NPAGEFORMAT HEADINGTYPE HEADINGX HEADINGY HEADINGS HEADINGINVALID
  STARTINGPAGE# FOLIOFORMAT FOLIOPRETEXT FOLIOPOSTTEXT PAGEOPTIONS NFPAGEFORMAT PAPERSIZE LANDSCAPE?)
  (SETQ SCRATCHSEL (fetch (TEXTOBJ SCRATCHSEL) of TEXTOBJ))
  [SETQ CH# (ADD1 (CDR (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#)
    ; Skip the SHOW button.
  (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
  (SETQ CH# (ADD1 (CDR NEXTB)))
  (SELECTQ (IMAGEOBJPROP (CAR NEXTB)
    'STATE)
    (|First(&Default)|
      (SETQ PAGEID 'FIRST))
    (Other% Left (SETQ PAGEID 'LEFT))
    (Other% Right (SETQ PAGEID 'RIGHT))
    (PROGN (TEDIT.PROMPTPRINT MAINTTEXT "Set KIND OF PAGE before APPLYing." T)
      (RETURN)))
    ; Find which page, for later.
  (SETQ STARTINGPAGE# (AND (EQ PAGEID 'FIRST)
    (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ CH#)))
  (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
  (SETQ CH# (ADD1 (CDR NEXTB)))
  (SETQ PAPERSIZE (OR (IMAGEOBJPROP (CAR NEXTB)
    'STATE)
    'Letter))
    ; Get the size of paper this is to be formatted for
  (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
  (SETQ CH# (ADD1 (CDR NEXTB)))
  (SETQ LANDSCAPE? (EQ (IMAGEOBJPROP (CAR NEXTB)
    'STATE)
    'ON))
    ; Decide if this kind of page is to be printed landscape....
  (SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ CH#))
  (SETQ CH# (ADD1 (CDR NEXTB)))
  (SELECTQ (IMAGEOBJPROP (CAR NEXTB)
    'STATE)
    (No (SETQ PAGENOS NIL))
    (Yes (SETQ PAGENOS T))
    NIL)
    ; Find about page numbers
  (SETQ PX (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ CH#))
  [SETQ PY (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
  [COND
    (PAGENOS
      ; If he wants page numbers, make sure he said WHERE to put
      ; them.
    (COND
      ((AND PX PY)
        (T (TEDIT.PROMPTPRINT MAINTTEXT "Please set the X and Y location for page numbers before
          APPLYing." T)
          (TEDIT.PROMPTFLASH MAINTTEXT)
          (RETURN)
        [SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
          ; Get to the numbering-format button
        (SETQ BUTTON (CAR NEXTB))
        (SETQ FOLIOFORMAT (SELECTQ (IMAGEOBJPROP BUTTON 'STATE)
          (123
            ; arabic numbers
          'ARABIC)
          (xiv
            ; lower-case roman numerals
          'LOWERROMAN)
          (XIV
            ; Upper-case roman numerals
          'UPPERROMAN)
          'ARABIC))
        [SETQ NEXTB (MBUTTON.FIND.NEXT.BUTTON TEXTOBJ (ADD1 (CDR NEXTB)
          ; Get to the number alignment button
        (SETQ BUTTON (CAR NEXTB))
        [SETQ ALIGNMENT (U-CASE (IMAGEOBJPROP BUTTON 'STATE)
          ; PX PY PFONT ALIGNMENT
          ; Margins: LEFT, RIGHT, TOP, BOTTOM
        (SETQ CH# (ADD1 (CDR NEXTB)))
        (SETQ FOLIOPRETEXT (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ CH#))
        [SETQ FOLIOPOSTTEXT (MBUTTON.NEXT.FIELD.AS.TEXT TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
```

::: Now get the margins on the paper

```
[SETQ LEFT (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
[SETQ RIGHT (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
[SETQ TOP (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
```



```
[SETQ BOTTOM (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
(COND
  [(SETQ COLS (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
    (T (TEDIT.PROMPTPRINT MAINTEXT "Please specify how many columns there should be." T)
      (TEDIT.PROMPTFLASH MAINTEXT)))
[SETQ COLWIDTH (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
[SETQ INTERCOL (MBUTTON.NEXT.FIELD.AS.NUMBER TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL)
; Col count, width, spacing
(SETQ HEADINGS (for HEADING# from 1 to 8 when (PROG1 [SETQ HEADINGTYPE
(MBUTTON.NEXT.FIELD.AS.ATOM
TEXTOBJ
(ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
[SETQ HEADINGX (MBUTTON.NEXT.FIELD.AS.NUMBER
TEXTOBJ
(ADD1 (fetch (SELECTION CH#)
of SCRATCHSEL]
[SETQ HEADINGY (MBUTTON.NEXT.FIELD.AS.NUMBER
TEXTOBJ
(ADD1 (fetch (SELECTION CH#)
of SCRATCHSEL])

collect (COND
  ((AND HEADINGX HEADINGY))
  (T (TEDIT.PROMPTPRINT MAINTEXT (CONCAT "You need to say WHERE " HEADINGTYPE
" headings go."))
    T)
    (TEDIT.PROMPTFLASH MAINTEXT)
    (SETQ HEADINGINVALID T)))
  (LIST HEADINGTYPE HEADINGX HEADINGY)))
(COND
  (HEADINGINVALID ; Headings invalid.
    (RETURN)))
[SETQ PFONT (\TEDIT.PARSE.CHARLOOKS.MENU TEXTOBJ (ADD1 (fetch (SELECTION CH#) of SCRATCHSEL]
```

;;; Glom all the oddball options (starting page, folio format &c) together

```
(SETQ PAGEOPTIONS (AND STARTINGPAGE# (LIST 'STARTINGPAGE# STARTINGPAGE#)))
(push PAGEOPTIONS (LIST FOLIOFORMAT FOLIOPRETEXT FOLIOPOSTTEXT))
(push PAGEOPTIONS 'FOLIOINFO)
(COND
  (LANDSCAPE? ; The pages are to be printed landscape. Remember that fact.
    (push PAGEOPTIONS T)
    (push PAGEOPTIONS 'LANDSCAPE?])
(SETQ NPAGEFORMAT (TEDIT.SINGLE.PAGEFORMAT PAGENOS PX PY PFONT (AND (NEQ ALIGNMENT 'OFF)
ALIGNMENT)
LEFT RIGHT TOP BOTTOM COLS COLWIDTH INTERCOL HEADINGS 'PICAS PAGEOPTIONS
PAPERSIZE))
(SETQ OPAGEFRAMES (fetch (TEXTOBJ TXTPAGEFRAMES) of MAINTTEXT))
(COND
  ((NOT (LISTP OPAGEFRAMES))
    (COND
      ((EQ PAGEID 'FIRST) ; Setting the first page sets them all
        (SETQ PAGEOPTIONS (COPY PAGEOPTIONS))
        (LISTPUT PAGEOPTIONS 'STARTINGPAGE# NIL) ; Starting page nubmer makes no sense on other than first
        ; pages.
        (SETQ NFPAGEFORMAT (TEDIT.SINGLE.PAGEFORMAT PAGENOS PX PY PFONT (AND (NEQ ALIGNMENT
'OFF)
ALIGNMENT)
LEFT RIGHT TOP BOTTOM COLS COLWIDTH INTERCOL HEADINGS 'PICAS
PAGEOPTIONS PAPERSIZE))
        (SETQ OPAGEFRAMES (LIST NPAGEFORMAT NFPAGEFORMAT NFPAGEFORMAT)))
      (T ; Otherwise, start from the default page layout
        (SETQ OPAGEFRAMES (COPY TEDIT.PAGE.FRAMES))
(SETQ PAGEID
  (FIRST (RPLACA OPAGEFRAMES NPAGEFORMAT))
  (LEFT (RPLACA (CDR OPAGEFRAMES)
NPAGEFORMAT))
  (RIGHT (RPLACA (CDDR OPAGEFRAMES)
NPAGEFORMAT))
  NIL)
(TEDIT.PAGEFORMAT MAINTTEXT OPAGEFRAMES)
(replace (TEXTOBJ \DIRTY) of MAINTTEXT with T) ; Mark the document as having changed.
(TTY.PROCESS (WINDOWPROP (WINDOWPROP W 'MAINWINDOW)
'PROCESS]))
```

(TEDIT.UNPARSE.PAGEFORMAT

[LAMBDA (PAGEREGION UNITS)

; Edited 12-Jun-90 18:59 by mitani

;;; Take a page layout and unparse it into a PList of specs.

```
(LET* ((PAPER (fetch (PAGEREGION REGIONSPEC) of PAGEREGION))
(PAPERWIDTH (fetch (REGION WIDTH) of PAPER))
(PAPERHEIGHT (fetch (REGION HEIGHT) of PAPER))
(REGIONS (fetch (PAGEREGION REGIONSUBBOXES) of PAGEREGION))
PX PY PFONT PQUAD PINFO LEFT RIGHT TOP BOTTOM (COLS 0)
COLWIDTH
(INTERCOL 0)
```

SPECS PAGENOS (OLDRIGHT NIL)
SCALEFACTOR HEADINGS
[for REGION in REGIONS do ;; Run thru the regions on the page, calculating information about the page as a whole.

```
(COND
  ((EQ (fetch (PAGEREGION REGIONFILLMETHOD) of REGION)
    'FOLIO) ; A page-number ("Folio") region
    (SETQ PAGENOS T)
    (SETQ PX (fetch (REGION LEFT) of (fetch REGIONSPEC of REGION)))
    (SETQ PY (fetch (REGION BOTTOM) of (fetch REGIONSPEC of REGION)))
    (SETQ SPECS (fetch REGIONLOCALINFO of REGION))
    (SETQ PFONT (LISTGET SPECS 'CHARLOOKS))
    [SETQ PQUAD (CADR (LISTGET SPECS 'PARALOOKS))
    (SELECTQ PQUAD
      (LEFT)
      (RIGHT (SETQ PX (IPLUS PX 288)))
      (CENTERED (SETQ PX (IPLUS PX 144)))
      NIL))
    [(EQ (fetch (PAGEREGION REGIONFILLMETHOD) of REGION)
      'HEADING) ; A page-heading region
      (SETQ HEADINGS (NCONC1 HEADINGS (LIST (LISTGET (fetch REGIONLOCALINFO
        of REGION)
        'HEADINGTYPE)
        (fetch (REGION LEFT)
          of (fetch REGIONSPEC of REGION))
        (fetch (REGION BOTTOM)
          of (fetch REGIONSPEC of REGION))
        (fetch REGIONSPEC of REGION))
        ; A regular-text region.
      (T
        (add COLS 1) ; Count columns
        (SETQ COLWIDTH (fetch (REGION WIDTH) of (fetch REGIONSPEC of REGION)))
        [SETQ RIGHT (IDIFFERENCE PAPERWIDTH (ADD1 (fetch (REGION RIGHT)
          of (fetch REGIONSPEC
            of REGION)]
          (COND
            ((EQ OLDRIGHT T))
            (OLDRIGHT (SETQ INTERCOL (IDIFFERENCE (fetch (REGION LEFT)
              of (fetch REGIONSPEC
                of REGION))
              OLDRIGHT))
              (SETQ OLDRIGHT T))
            (T (SETQ OLDRIGHT (fetch (REGION RIGHT) of (fetch REGIONSPEC
              of REGION)))
              (SETQ LEFT (fetch (REGION LEFT) of (fetch REGIONSPEC of REGION)))
              [SETQ TOP (IDIFFERENCE PAPERHEIGHT (fetch (REGION PTOP)
                of (fetch REGIONSPEC
                  of REGION)]
                (SETQ BOTTOM (fetch (REGION BOTTOM) of (fetch REGIONSPEC
                  of REGION]
```

(SELECTQ UNITS
 ((POINTS NIL) (* If units are in printers points, the default, do no scaling)
)
 (PICAS (* The units are in picas--12pts per.
 Scale all values.)
 (SETQ SCALEFACTOR 0.12))
 (INCHES (* The units are in inches, at 72.27pts per.
 Set the scale factor)
 (SETQ SCALEFACTOR 0.7227))
 (CM (* Units are in CM, at 72.27/2.54pts per.)
 (SETQ SCALEFACTOR (CONSTANT (FQUOTIENT 0.7227 2.54))))
 (\ILLEGAL.ARG UNITS))

```
[COND
  (SCALEFACTOR (* We need to do the scaling.)
    (AND PX (SETQ PX (FQUOTIENT (FIXR (FQUOTIENT PX SCALEFACTOR)
      100)))
    (AND PY (SETQ PY (FQUOTIENT (FIXR (FQUOTIENT PY SCALEFACTOR)
      100)))
    (AND LEFT (SETQ LEFT (FQUOTIENT (FIXR (FQUOTIENT LEFT SCALEFACTOR)
      100)))
    (AND RIGHT (SETQ RIGHT (FQUOTIENT (FIXR (FQUOTIENT RIGHT SCALEFACTOR)
      100)))
    (AND TOP (SETQ TOP (FQUOTIENT (FIXR (FQUOTIENT TOP SCALEFACTOR)
      100)))
    (AND BOTTOM (SETQ BOTTOM (FQUOTIENT (FIXR (FQUOTIENT BOTTOM SCALEFACTOR)
      100)))
    (AND COLWIDTH (SETQ COLWIDTH (FQUOTIENT (FIXR (FQUOTIENT COLWIDTH SCALEFACTOR)
      100)))
    (AND INTERCOL (SETQ INTERCOL (FQUOTIENT (FIXR (FQUOTIENT INTERCOL SCALEFACTOR)
      100)))
    (SETQ HEADINGS (for HDG in HEADINGS collect (LIST (CAR HDG)
      (FQUOTIENT (FIXR (FQUOTIENT (CADR HDG)
        SCALEFACTOR))
        100)
      (FQUOTIENT (FIXR (FQUOTIENT (CADDR HDG)
        SCALEFACTOR))
        100)
      100]
```

(LIST PAGENOS PX PY PFONT PQUAD LEFT RIGHT TOP BOTTOM COLS COLWIDTH INTERCOL HEADINGS
(fetch (PAGEREGION REGIONLOCALINFO) of PAGEREGION])

)

:: Initialization Code

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS TEDIT.EXPANDED.MENU TEDIT.EXPANDEDPARA.MENU TEDIT.CHARLOOKS.MENU TEDIT.MENUDIVIDER.SPEC
TEDIT.EXPANDEDMENU.SPEC TEDIT.CHARLOOKSMENU.SPEC TEDIT.PARAMENU.SPEC TEDIT.PAGEMENU.SPEC
TEDIT.EXPANDED.PAGEMENU)

)

(DEFINEQ

(\TEDIT.MENU.INIT

[LAMBDA NIL

; Edited 30-Mar-94 15:53 by jds

::: Initialize the descriptions for all TEdit menus

::: Divides between the main page layout menu and page-# font submenu

(SETQ TEDIT.MENUDIVIDER.SPEC (LIST (create MB.TEXT
MBSTRING _ "
")))

::: The principal expanded menu

(SETQ TEDIT.EXPANDEDMENU.SPEC (LIST (create MB.BUTTON
MBLABEL _ "Quit")
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "Page Layout")
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "Char Looks")
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "Para Looks")
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "All")
(create MB.TEXT
MBSTRING _ " ")
(create MB.TOGGLE
MBTEXT _ "Unformatted"
MBCHANGESTATEFN _ (FUNCTION \TEDITMENU.RECORD.UNFORMATTED))
(create MB.TEXT
MBSTRING _ "
")
(create MB.BUTTON
MBLABEL _ "Get")
(create MB.INSERT)
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "Put")
(create MB.INSERT)
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "Include")
(create MB.INSERT)
(create MB.TEXT
MBSTRING _ "
")
(create MB.BUTTON
MBLABEL _ "Find")
(create MB.INSERT)
(create MB.TEXT
MBSTRING _ " ")
(create MB.BUTTON
MBLABEL _ "Substitute")
(create MB.INSERT)
(create MB.TEXT
MBSTRING _ " for")
(create MB.INSERT)
(create MB.TEXT
MBSTRING _ " ")
(create MB.TOGGLE
MBTEXT _ "Confirm")
(create MB.TEXT
MBSTRING _ " ")

```

(create MB.TOGGLE
  MBTEXT _ "Use New Looks")
(create MB.TEXT
  MBSTRING _ "
  ")
(create MB.BUTTON
  MBLABEL _ "Hardcopy")
(create MB.TEXT
  MBSTRING _ " server:")
(create MB.INSERT)
(create MB.TEXT
  MBSTRING _ " copies:")
(create MB.INSERT)
(create MB.TEXT
  MBSTRING _ "
  ")
(create MB.TEXT
  MBSTRING _ "Print ")
(create MB.NWAY
  MBBUTTONS _ '(One% Side Duplex)
  MBMAXITEMSPERLINE _ 5)
(create MB.TEXT
  MBSTRING _ " Message/Phone#:")
(create MB.INSERT))

```

::: The character-looks (font, etc.) menu

```

(SETQ TEDIT.CHARLOOKSMENU.SPEC (LIST (create MB.TEXT
  MBSTRING _ "Props: "
  MBFONT _ (FONTCREATE 'HELVETICA 8))
  (create MB.3STATE
  MBLABEL _ 'Bold)
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.3STATE
  MBLABEL _ 'Italic)
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.3STATE
  MBLABEL _ 'Underline)
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.3STATE
  MBLABEL _ 'StrikeThru)
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.3STATE
  MBLABEL _ 'Overbar)
  (create MB.TEXT
  MBSTRING _ "
  ")
  (create MB.NWAY
  MBBUTTONS _ '(TimesRoman Helvetica Gacha Modern Classic Terminal
  Other)
  MBMAXITEMSPERLINE _ 5)
  (create MB.TEXT
  MBSTRING _ "other font:")
  (create MB.INSERT)
  (create MB.TEXT
  MBSTRING _ "
  ")
  (create MB.TEXT
  MBSTRING _ "Size: "
  MBFONT _ (FONTCREATE 'HELVETICA 8))
  (create MB.INSERT)
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.NWAY
  MBBUTTONS _ '(Normal Superscript Subscript))
  (create MB.TEXT
  MBSTRING _ " distance: "
  MBFONT _ (FONTCREATE 'HELVETICA 8))
  (create MB.INSERT)))

```

::: The paragraph-formatting menu (margins, etc.)

```

(SETQ TEDIT.PARAMENU.SPEC (LIST (create MB.BUTTON
  MBLABEL _ 'APPLY
  MBBUTTONEVENTFN _ (FUNCTION \TEDIT.APPLY.PARALOOKS))
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.BUTTON
  MBLABEL _ 'SHOW
  MBBUTTONEVENTFN _ (FUNCTION \TEDIT.SHOW.PARALOOKS))
  (create MB.TEXT
  MBSTRING _ " ")
  (create MB.BUTTON

```

```

        MBLABEL _ 'NEUTRAL
        MBBUTTONEVENTFN _ (FUNCTION \TEDIT.NEUTRALIZE.PARALOOKS.MENU)
    (create MB.TEXT
        MBSTRING _ "
    )
    (create MB.NWAY
        MBBUTTONS _ '(Left Right Centered Justified)
    (create MB.TEXT
        MBSTRING _ " ")
    (create MB.3STATE
        MBLABEL _ "Page Heading")
    (create MB.TEXT
        MBSTRING _ " type:")
    (create MB.INSERT)
    (create MB.TEXT
        MBSTRING _ "
        Line leading:"
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.INSERT)
    (create MB.TEXT
        MBSTRING _ "pts Para Leading:"
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.INSERT)
    (create MB.TEXT
        MBSTRING _ "pts Special Locn: X"
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.INSERT)
    (create MB.TEXT
        MBSTRING _ "picas, Y"
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.INSERT)
    (create MB.TEXT
        MBSTRING _ "picas
        New Page: "
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.3STATE
        MBLABEL _ "Before")
    (create MB.TEXT
        MBSTRING _ " ")
    (create MB.3STATE
        MBLABEL _ "After")
    (create MB.TEXT
        MBSTRING _ " Display mode: "
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.3STATE
        MBLABEL _ "Hardcopy")
    (create MB.TEXT
        MBSTRING _ " Keep: "
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.3STATE
        MBLABEL _ "Heading")
    (create MB.TEXT
        MBSTRING _ "
        Tab Type: "
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.NWAY
        MBBUTTONS _ '((Left \TEDIT.TABTYPE.SET)
                    (Right \TEDIT.TABTYPE.SET)
                    (Centered \TEDIT.TABTYPE.SET)
                    (Decimal \TEDIT.TABTYPE.SET)

    (create MB.TEXT
        MBSTRING _ " ")
    (create MB.TOGGLE
        MBTEXT _ "Dotted Leader"
        MBCHANGESTATEFN _ (FUNCTION \TEDIT.RECORD.TABLEADERS))
    (create MB.TEXT
        MBSTRING _ " Default Tab Size:"
        MBFONT _ (FONTCREATE 'HELVETICA 8))
    (create MB.INSERT)
    (create MB.TEXT
        MBSTRING _ "
    )
    (create MB.MARGINBAR)
    (create MB.TEXT
        MBSTRING _ "
    )))

```

;;; Page-layout menu for columns, page headings, page numbers, etc.

```

    (SETQ TEDIT.PAGEMENU.SPEC (APPEND (LIST (create MB.BUTTON
        MBLABEL _ 'APPLY
        MBBUTTONEVENTFN _ '\TEDIT.APPLY.PAGEFORMATTING)
    (create MB.TEXT
        MBSTRING _ " "
        MBFONT _ (FONTCREATE 'HELVETICA 8 'BOLD))
    (create MB.BUTTON
        MBLABEL _ 'SHOW

```

```

        MBBUTTONEVENTFN _ '\TEDIT.SHOW.PAGEFORMATTING)
(create MB.TEXT
        MBSTRING _ "
        ")
(create MB.TEXT
        MBSTRING _ "For page: ")
(create MB.NWAY
        MBBUTTONS _ '( |First(&Default) | Other% Left Other% Right))
(create MB.TEXT
        MBSTRING _ "
        Starting Page #: ")
(create MB.INSERT
        MBINITENTRY _ 1)
(create MB.TEXT
        MBSTRING _ " Paper Size: ")
(create MB.NWAY
        MBBUTTONS _ '(Letter Legal A4)
        MBINITSTATE _ 'Letter)
(create MB.TEXT
        MBSTRING _ " ")
(create MB.TOGGLE
        MBTEXT _ "Landscape")
(create MB.TEXT
        MBSTRING _ "
        ")
(create MB.TEXT
        MBSTRING _ "Page numbers: ")
(create MB.TEXT
        MBSTRING _ " ")
        MBFONT _ (FONTCREATE 'HELVETICA 8 'BOLD))
(create MB.NWAY
        MBBUTTONS _ '(No Yes)
        MBINITSTATE _ 'Yes)
(create MB.TEXT
        MBSTRING _ " ")
(create MB.TEXT
        MBSTRING _ "X: ")
(create MB.INSERT
        MBINITENTRY _ 25.5)
(create MB.TEXT
        MBSTRING _ " ")
(create MB.TEXT
        MBSTRING _ "Y: ")
(create MB.INSERT
        MBINITENTRY _ 3)
(create MB.TEXT
        MBSTRING _ " Format: ")
(create MB.NWAY
        MBBUTTONS _ '(123 xiv XIV)
        MBINITSTATE _ '123)
(create MB.TEXT
        MBSTRING _ " ")
(create MB.TEXT
        MBSTRING _ "Alignment: ")
(create MB.NWAY
        MBBUTTONS _ '(Left Centered Right)
        MBINITSTATE _ 'Centered)
(create MB.TEXT
        MBSTRING _ "
        ")
(create MB.TEXT
        MBSTRING _ " Text before number: ")
(create MB.INSERT
        MBINITENTRY _ "")
(create MB.TEXT
        MBSTRING _ " Text after number: ")
(create MB.INSERT
        MBINITENTRY _ "")
(create MB.TEXT
        MBSTRING _ "
        ")
(LIST (create MB.TEXT
        MBSTRING _ "Margins: Left")
(create MB.INSERT
        MBINITENTRY _ 6)
(create MB.TEXT
        MBSTRING _ " Right")
(create MB.INSERT
        MBINITENTRY _ 6)
(create MB.TEXT
        MBSTRING _ " Top")
(create MB.INSERT
        MBINITENTRY _ 6)
(create MB.TEXT
        MBSTRING _ " Bottom")

```



```
        MBSTRING _ " Y:")
(create MB.INSERT)
(create MB.TEXT
        MBSTRING _ " Heading Type:")
(create MB.INSERT)
(create MB.TEXT
        MBSTRING _ " X:")
(create MB.INSERT)
(create MB.TEXT
        MBSTRING _ " Y:")
(create MB.INSERT])
```

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\TEDIT.MENU.INIT)

(\TEDITMENU.CREATE)

(\TEDIT.CHARLOOKSMENU.CREATE)

(\TEDITPARAMENU.CREATE)

(\TEDITPAGEMENU.CREATE)

)

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

(ADDTOVAR NLAMA)

(ADDTOVAR NLAML)

(ADDTOVAR LAMA)

)

(PUTPROPS TEDITMENU COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1985 1986 1987 1988 1989 1990 1991 1992
1993 1994 1995))

FUNCTION INDEX

DRAWMARGINSCALE	16	MB.THREESTATE.WHENOPERATEDFN	9	\TEDIT.CHARLOOKSMENU.CREATE	29
MARGINBAR	16	MB.THREESTATEBUTTON.FN	9	\TEDIT.EXPANDED.MENU	26
MARGINBAR.CREATE	18	MB.WHENOPERATEDFN	3	\TEDIT.EXPANDEDCHARLOOKS.MENU	29
MARGINBAR.INIT	22	MBUTTON.CHANGENAME	4	\TEDIT.EXPANDEDPARA.MENU	33
MB.BUTTONEVENTINFN	2	MBUTTON.CREATE	4	\TEDIT.FILL.IN.CHARLOOKS.MENU	30
MB.COPYFN	3	MBUTTON.FIND.BUTTON	4	\TEDIT.MENU.INIT	43
MB.CREATE.NWAYBUTTON	10	MBUTTON.FIND.NEXT.BUTTON	5	\TEDIT.NEUTRALIZE.CHARLOOKS	30
MB.CREATE.THREESTATEBUTTON	7	MBUTTON.FIND.NEXT.FIELD	5	\TEDIT.NEUTRALIZE.CHARLOOKS.MENU	31
MB.DEFAULTBUTTON.ACTIONFN	27	MBUTTON.INIT	5	\TEDIT.NEUTRALIZE.PARALOOKS.MENU	36
MB.DEFAULTBUTTON.FN	26	MBUTTON.NEXT.FIELD.AS.ATOM	6	\TEDIT.PARSE.CHARLOOKS.MENU	31
MB.DISPLAY	2	MBUTTON.NEXT.FIELD.AS.NUMBER	6	\TEDIT.RECORD.TABLEADERS	37
MB.GETFN	3	MBUTTON.NEXT.FIELD.AS.PIECES	6	\TEDIT.SHOW.CHARLOOKS	29
MB.MARGINBAR.DISPLAYFN	21	MBUTTON.NEXT.FIELD.AS.TEXT	6	\TEDIT.SHOW.PAGEFORMATTING	37
MB.MARGINBAR.SELFN	18	MBUTTON.SET.FIELD	6	\TEDIT.SHOW.PARALOOKS	35
MB.MARGINBAR.SHOWTAB	21	MBUTTON.SET.NEXT.BUTTON.STATE	7	\TEDIT.TABTYPE.SET	22
MB.MARGINBAR.SIZEFN	20	MBUTTON.SET.NEXT.FIELD	6	\TEDITMENU.CREATE	26
MB.MARGINBAR.TABTRACK	22	MDESCALE	21	\TEDITMENU.RECORD.UNFORMATTED	27
MB.NB.DISPLAYFN	10	MSCALE	21	\TEDITMENU.SELSCREENER	7
MB.NB.PACKITEMS	12	NWAYBUTTON.INIT	12	\TEDITPAGEMENU.CREATE	39
MB.NB.SIZEFN	11	TEDIT.UNPARSE.PAGEFORMAT	41	\TEDITPARAMENU.CREATE	33
MB.NB.WHENOPERATEDFN	11	TEDITMENU.STREAM	7	\TEXTMENU.DOC.CREATE	23
MB.NWAYBUTTON.ADDITEM	13	TEXTMENU.CLOSEFN	25	\TEXTMENU.SET.TOGGLE	15
MB.NWAYBUTTON.SELFN	11	THREESTATE.INIT	9	\TEXTMENU.START	23
MB.NWAYMENU.NEWBUTTON	12	\TEDIT.APPLY.BOLDNESS	29	\TEXTMENU.TOGGLE.CREATE	13
MB.PUTFN	4	\TEDIT.APPLY.CHARLOOKS	29	\TEXTMENU.TOGGLE.DISPLAY	14
MB.SELFN	3	\TEDIT.APPLY.OLINE	29	\TEXTMENU.TOGGLE.INIT	15
MB.SETIMAGE	2	\TEDIT.APPLY.PAGEFORMATTING	40	\TEXTMENU.TOGGLE.SHOWSELFN	14
MB.SHOWSELFN	4	\TEDIT.APPLY.PARALOOKS	33	\TEXTMENU.TOGGLE.WHENOPERATEDFN	15
MB.SIZEFN	3	\TEDIT.APPLY.SLOPE	32	\TEXTMENU.TOGGLEFN	15
MB.THREESTATE.DISPLAY	8	\TEDIT.APPLY.STRIKEOUT	32		
MB.THREESTATE.SHOWSELFN	8	\TEDIT.APPLY.ULINE	33		

VARIABLE INDEX

IMAGEOBJTYPES	7,13,16,23	\TEDIT.CENTERTAB	23	\TEDIT.DOTTED.LEFTTAB	23
TEDIT.EXTENDEDRIGHTMARK	23	\TEDIT.DECIMALTAB	23	\TEDIT.DOTTED.RIGHTTAB	23
TEXTMENUICON	26	\TEDIT.DOTTED.CENTERTAB	23	\TEDIT.LEFTTAB	23
TEXTMENUICONMASK	26	\TEDIT.DOTTED.DECIMALTAB	23	\TEDIT.RIGHTTAB	23

CONSTANT INDEX

\SCRATCHLEN	2
-------------------	---