File created: 16-May-90 11:59:31 **{DSK}<usr>local>lde>lispcore>sources>AFONT.;2**

changes to: (VARS AFONTCOMS)

previous date: 14-Sep-87 11:59:36 {DSK}<usr>local>lde>lispcore>sources>AFONT.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS


```
(RPAQQ AFONTCOMS
       ((XCL:FILE-ENVIRONMENTS "AFONT")
        (DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS BOUNDINGBOX FONTBOUNDINGBOX)
               (CONSTANTS noInfoCode))
        (FNS \CREATESTARFONT \READACFONTBOXES \READACFONTFILE \ACCHARIMAGELIST \ACCHARWIDTHLIST \GETFBB
             \ACCHARPOSLIST \ACROTATECHAR \READFONTWDFILE \FACECODE \FAMILYCODE \FINDFONT)
        [INITVARS (INTERPRESSFONTDIRECTORIES '("{Erinyes}<Lyric>Fonts>"]
        (MACROS \POSITIONFONTFILE)))


(XCL:DEFINE-FILE-ENVIRONMENT "AFONT" :PACKAGE "IL"
   :READTABLE "INTERLISP"
   :COMPILER :COMPILE-FILE)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD BOUNDINGBOX (

          (* * The bounding box for a character in an AC file)

                    BBOX                                    (* Offset from the left edge of the bounding box to the
                                                            character's origin)
                    BBOY                                    (* Offset from the bottom of the bounding box to the character's
                                                            origin)
                    BBDX                                    (* Width of the character's bounding box in pixels)
                    BBDY                                    (* Height of the bounding box in bits;
                                                            -1 if this character doesn't really exist)
                    RASTERWIDTHX                            (* Width of the character's image
                                                            (i.e., the escapement for this character) in raster bits)
                    RASTERWIDTHY                            (* Amount this char moves in Y, in raster units.)
                    ))

(RECORD FONTBOUNDINGBOX (FBBBDX FBBBDY FBBBOX FBBBOY))
)

(DECLARE%: EVAL@COMPILE

(RPAQQ noInfoCode 32768)

(CONSTANTS noInfoCode)
)
)

(DEFINEQ

(\CREATESTARFONT
  [LAMBDA (FAMILY PSIZE FACE ROTATION DEVICE CHARSET)          (* gbn " 1-Oct-85 18:29")
    ;; the Build font descriptor for an Interpress NS font.  If we can't find widths info for that font, return NIL

    ;; Widths array is fully allocated, with zeroes for characters with no information.  An array is not allocated for fixed WidthsY.  DEVICE is PRESS or
    ;; INTERPRESS

    (DECLARE (GLOBALVARS INTERPRESSFONTDIRECTORIES \ASCIITONS))
    (RESETLST                                                  ; RESETLST to make sure the fontfiles get closed
        (PROG [(CS (OR CHARSET \DEFAULTCHARSET))
               (NSMICASIZE (FIXR (FQUOTIENT (ITIMES PSIZE 2540)
                                            72)))
               (FD (create FONTDESCRIPTOR
                           FONTDEVICE _ DEVICE
                           FONTFAMILY _ FAMILY
                           FONTSIZE _ PSIZE
                           FONTFACE _ FACE
                           \SFFACECODE _ (\FACECODE FACE)
                           ROTATION _ ROTATION
                           OTHERDEVICEFONTPROPS _ \ASCIITONS
                           FONTSCALE _ (CONSTANT (FQUOTIENT 2540 72]
              (RETURN (if (NOT (\GETCHARSETINFO CS FD T))
                          then                                 ; return NIL and let FONTCREATE decide whether or not to
                                                               ; cause an error
                              NIL
```

```
                        else FD)))))])
```

(\READACFONTBOXES

```
  [LAMBDA (FILE STARTCHAR ENDCHAR)                                  (* jds "15-Jun-85 11:48")
                                                                    ; GETACCHARSPECS returns (bbox bboy bbdx bbdy)
                                                                    ; if bbdx and bbdy are both zero, then treat it as a space.
    (SETFILEPTR FILE 48)                                            ; Move to the start of AC file's width info.
    (for X from STARTCHAR to ENDCHAR collect                        ; Now collect the 4 bounding box values into a list
                                          (create BOUNDINGBOX
                                                  RASTERWIDTHX _ (PROG1 (\WIN FILE)
                                                                    ; Read a fraction, and truncate it to an integer # of raster bits
                                                                        (\WIN FILE))
                                                  RASTERWIDTHY _ (PROG1 (\WIN FILE)
                                                                    ; Read a fraction, and truncate it to an integer # of raster bits
                                                                        (\WIN FILE))
                                                  BBOX _ (SIGNED (\WIN FILE)
                                                                 BITSPERWORD)
                                                  BBOY _ (SIGNED (\WIN FILE)
                                                                 BITSPERWORD)
                                                  BBDX _ (SIGNED (\WIN FILE)
                                                                 BITSPERWORD)
                                                  BBDY _ (SIGNED (\WIN FILE)
                                                                 BITSPERWORD])
```

(\READACFONTFILE

```
  [LAMBDA (STRM FAMILY SIZE FACE PAD.LEFT DONT.PAD.RIGHT)           ; Edited 1-Sep-87 10:04 by Snow
    ;; Read an AC-format font file.  Assumes that the file is open and has already been determined to be of type AC.
    [COND
       ((RANDACCESSP STRM)
        (RESETSAVE NIL (LIST (FUNCTION CLOSEF?)
                             STRM)))
       (T
          ;; This is necessary unless we figure out how to read the AC file sequentially.  When we figure this out, we can factor the RESETSAVE
          ;; back in \READDISPLAYFONTFILE
          (SETQ STRM (OPENSTREAM (CLOSEF? STRM)
                                 'INPUT))
          (RESETSAVE NIL (LIST (FUNCTION CLOSEF?)
                               STRM))
          (COPYBYTES STRM (SETQ STRM (OPENSTREAM '{NODIRCORE} 'BOTH]
    (SETFILEPTR STRM 28)                                            ; Starting at 28 skips the family and face bytes.
    (PROG [FBBLIST STARTCHAR ENDCHAR CHARWIDTHLIST CHARIMAGEWIDTHLIST LEFTKERNS OFFSETS WIDTHS IMAGEWIDTHS
           FONTDESC FBBBITMAP CHARBITMAP STARTWORDLIST BBOXLIST DUMMYCHAROFFSET DUMMYWIDTH
           (CSINFO (create CHARSETINFO
                           IMAGEWIDTHS _ (\CREATECSINFOELEMENT)
                           LEFTKERN _ (\CREATEKERNELEMENT]
          (SETQ STARTCHAR (BIN STRM))                               ; Get the first and last characters in this font
          (SETQ ENDCHAR (BIN STRM))
          (SETQ BBOXLIST (\READACFONTBOXES STRM STARTCHAR ENDCHAR))
                                                                    ; Read the list of bounding boxes for all the chars in the font
          (SETQ FBBLIST (\GETFBB BBOXLIST))
          (SETQ CHARWIDTHLIST (\ACCHARIMAGELIST BBOXLIST))          ; And the escapement for each character.
          (SETQ CHARIMAGEWIDTHLIST (\ACCHARWIDTHLIST BBOXLIST FBBLIST))
                                                                    ; Create the list of character widths for the characters in the font.
          (COND
             ([EVERY (CDR CHARWIDTHLIST)
                     (FUNCTION (LAMBDA (WID)
                                 (OR (ZEROP WID)
                                     (EQP WID (CAR CHARWIDTHLIST]
                                                                    ; Fixed-pitch font.  Make the dummy character (for non-existent
                                                                    ; chars) the same width.
              (SETQ DUMMYWIDTH (CAR CHARWIDTHLIST)))
             (T                                                     ; Otherwise, make the dummy 6 wide.
                (SETQ DUMMYWIDTH 6)))
          (COND
             ((NULL (REMOVE 0 CHARIMAGEWIDTHLIST))
              (ERROR "No raster images" NIL)
              (RETURN)))
          (SETQ LEFTKERNS (FETCH (CHARSETINFO LEFTKERN) OF CSINFO))
          (FOR I FROM STARTCHAR TO ENDCHAR AS BOX IN BBOXLIST DO
                                                                    ; set the left kerning values.  the default value is ZERO which is
                                                                    ; set when the element is created.  Currently it is an array
                                                                    ; because kerning values can be negative values.
                                                                       (\FSETLEFTKERN LEFTKERNS I
                                                                              (FFETCH (BOUNDINGBOX BBOX)
                                                                                 OF BOX)))
          (SETQ IMAGEWIDTHS (fetch (CHARSETINFO IMAGEWIDTHS) of CSINFO))
          (for I from 0 to (ADD1 \MAXTHINCHAR) do (\FSETIMAGEWIDTH IMAGEWIDTHS I DUMMYWIDTH))
          (SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
          (for I from 0 to (ADD1 \MAXTHINCHAR) do (\FSETWIDTH WIDTHS I DUMMYWIDTH))
                                                                    (* SETQ IMAGEWIDTHS (ARRAY 258
                                                                    (QUOTE (BITS 16)) DUMMYWIDTH 0))
    ;; Create the array of character widths, assuming the dummy width for all characters--we'll write over it later
          [for X from STARTCHAR to ENDCHAR as Y in CHARIMAGEWIDTHLIST
```

```
   do ;; Fill in the image widths (the width of the image, as against how far to space over after printing the character)
      (\FSETIMAGEWIDTH IMAGEWIDTHS X (COND
                                        ((ZEROP Y)
                                         0)
                                        (T (IPLUS Y (COND
                                                       (PAD.LEFT 1)
                                                       (T 0))
                                                   (COND
                                                      (DONT.PAD.RIGHT 0)
                                                      (T 1]
                                         ; And the array of image escapements
(for X from STARTCHAR to ENDCHAR as Y in CHARWIDTHLIST do (\FSETWIDTH WIDTHS X Y))
[replace CHARSETDESCENT of CSINFO with (IMAX 0 (IMINUS (fetch (FONTBOUNDINGBOX FBBBOY) of FBBLIST]
[replace CHARSETASCENT of CSINFO with (IMAX 0 (IPLUS (fetch (FONTBOUNDINGBOX FBBBDY) of FBBLIST)
                                                     (fetch (FONTBOUNDINGBOX FBBBOY) of FBBLIST]
[replace CHARSETBITMAP of CSINFO with (SETQ CHARBITMAP (BITMAPCREATE (IPLUS (SETQ DUMMYCHAROFFSET
                                                                               (for (X _ STARTCHAR)
                                                                                  to ENDCHAR
                                                                                  sum (\FGETWIDTH
                                                                                            IMAGEWIDTHS X
                                                                                            )))
                                                                             DUMMYWIDTH)
                                                                      (fetch (FONTBOUNDINGBOX FBBBDY)
                                                                         of FBBLIST]
(SETQ OFFSETS (fetch (CHARSETINFO OFFSETS) of CSINFO))
(for I from 0 to (ADD1 \MAXTHINCHAR) do (\FSETOFFSET OFFSETS I DUMMYCHAROFFSET))
(SETQ STARTWORDLIST (\ACCHARPOSLIST STRM STARTCHAR ENDCHAR))
(bind (DESTLEFT _ 0) for NTHCHAR from STARTCHAR to ENDCHAR as BBLIST in BBOXLIST as STARTWORD
   in STARTWORDLIST as CHARWIDTH in CHARWIDTHLIST
   do (PROG (RASTERINFO BBOX BBBITMAP BBBMBASE)        ; \ACCHARPOSLIST returns NIL if no raster exists for the code
            (COND
               ((NULL STARTWORD)
                ;; This character has no image;  use the dummy char's offset (already in the offset and width arrays from earlier)
                (add DESTLEFT (\FGETWIDTH IMAGEWIDTHS NTHCHAR))
                (\FSETWIDTH WIDTHS NTHCHAR DUMMYWIDTH)
                (\FSETIMAGEWIDTH IMAGEWIDTHS NTHCHAR DUMMYWIDTH)
                (GO L2)))
            (SETFILEPTR STRM STARTWORD)               ; If could flush this, would work on non-randaccessp devices
            (SETQ RASTERINFO (\WIN STRM))
            (COND
               ((EQ -1 (fetch BBDY of BBLIST))
                (\FSETWIDTH WIDTHS NTHCHAR DUMMYWIDTH)
                (\FSETIMAGEWIDTH IMAGEWIDTHS NTHCHAR DUMMYWIDTH)
                (GO L2)))                             ; \ACCHARPOSLIST returns NIL if no raster exists for the code
            (SETQ BBOX (fetch BBOX of BBLIST))
            (COND
               ((AND (ZEROP (fetch BBDX of BBLIST))
                     (ZEROP (fetch BBDY of BBLIST)))  ; The image is zero wide or zero high.  Don't bother reading a
                                                      ; bitmap image

                )
               ((SETQ BBBITMAP (BITMAPCREATE (TIMES 16 (FOLDLO RASTERINFO 1024))
                                             (IMOD RASTERINFO 1024)))
                (SETQ BBBMBASE (fetch BITMAPBASE of BBBITMAP))
                ;; STARTWORD is the characters raster information word.  The high 6 bits record number of words per scan line and
                ;; the lower 10 bits is the same as bbdx bbdx.  The raster for the char follows STARTWORD
                (\BINS STRM BBBMBASE 0 (TIMES 2 (FOLDLO RASTERINFO 1024)
                                                (IMOD RASTERINFO 1024)))
                (SETQ BBBITMAP (\ACROTATECHAR BBBITMAP))
                                                      ; here is the place to add a rotation function to manipulate the
                                                      ; character images coming off *.ac
                (BITBLT BBBITMAP 0 0 CHARBITMAP [PLUS DESTLEFT (IMAX 0 (COND
                                                                          (PAD.LEFT (ADD1 BBOX))
                                                                          (T BBOX]
                        (DIFFERENCE (fetch BBOY of BBLIST)
                                    (fetch (FONTBOUNDINGBOX FBBBOY) of FBBLIST))
                        (\FGETWIDTH IMAGEWIDTHS NTHCHAR)
                        (CADDDR BBLIST)
                        'INPUT
                        'REPLACE)                     ; ADD1 to BBOX because we add an empty column to each
                                                      ; raster image to the left

                ))
            (\FSETOFFSET OFFSETS NTHCHAR DESTLEFT)
      ;; on screen ac fonts, there are no spaces stored so that the width of the char is exactly that of the character image without any
      ;; spacing columns
            (add DESTLEFT (\FGETWIDTH IMAGEWIDTHS NTHCHAR))
       L2                                             ; add 2 because of the two blank columns we add;  one on either
                                                      ; side of the ac raster image
      ))
(BITBLT NIL 0 0 CHARBITMAP (ADD1 DUMMYCHAROFFSET)
      0
      (IDIFFERENCE DUMMYWIDTH 2)
```

```
               NIL
               'TEXTURE
               'REPLACE BLACKSHADE)                          ; Fill in the dummy-character black blot
         (RETURN CSINFO])
```

(\ACCHARIMAGELIST

```
  [LAMBDA (BOXLIST)                                          (* jds "15-Jun-85 11:37")
    ;; Returns a list of the ESCAPEMENTS (ie how far to move after printng this character) for each char in the font.
    (for BOX in BOXLIST collect (fetch (BOUNDINGBOX RASTERWIDTHX) of BOX])
```

(\ACCHARWIDTHLIST

```
  [LAMBDA (BOXLIST FBBOX)                                    (* jds " 4-Dec-84 16:05")
                                                             ; GETACCHARSPECS returns (bbox bboy bbdx bbdy)
                                                             ; if bbdx and bbdy are both zero, then treat it as a space.
    (for BOX in BOXLIST bind (STARTWORD BBOX BBOY BBDX BBDY)
       collect (SETQ BBOX (fetch BBOX of BOX))
               (SETQ BBOY (fetch BBOY of BOX))
               (SETQ BBDX (fetch BBDX of BOX))
               (SETQ BBDY (fetch BBDY of BOX))
               (COND
                  ((AND (ZEROP BBDX)
                        (ZEROP BBDY))                        ; we've found a Space.  Smash in a quarter of the maximum
                                                             ; width.  Maybe should be an explicit em?
                   (IMAX 2 (FOLDLO (IPLUS 2 (fetch (FONTBOUNDINGBOX FBBBDX) of FBBOX))
                                   4)))
                  (T (COND
                        ((EQ BBDX -1)
                         0)
                        (T (IPLUS BBDX (IMAX 0 BBOX])
```

(\GETFBB

```
  [LAMBDA (BOXLIST)                                          (* jds "17-May-85 10:22")
                                                             ; Read a font bounding box from an AC file
    (PROG (RESULTLIST CHARCOUNT BBLIST MAXBBOX MAXBBOY MINBBOX MINBBOY MAXSUMBBOXBBDX MAXSUMBBOYBBDY BBOX BBOY
                 BBDX BBDY)                                  ; \GETFBB returns the fbbdx fbbdy fbbox fbboy of an acfont
          (SETQ MINBBOX 32767)
          (SETQ MINBBOY 32767)
          (SETQ MAXBBOX -32768)
          (SETQ MAXBBOY -32768)
          (SETQ MAXSUMBBOXBBDX -32768)
          (SETQ MAXSUMBBOYBBDY -32768)
          [for BOX in BOXLIST do (SETQ BBOX (fetch (BOUNDINGBOX BBOX) of BOX))
                                 (SETQ BBOY (fetch (BOUNDINGBOX BBOY) of BOX))
                                 (SETQ BBDX (fetch (BOUNDINGBOX BBDX) of BOX))
                                 (SETQ BBDY (fetch (BOUNDINGBOX BBDY) of BOX))
                                                             ; GETACCHARSPECS returns bbox bboy bbdx bbdy
                                 (COND
                                    [(IEQP BBDY -1)          ; This character doesn't exist.  Create a dummy bounding box for
                                                             ; it
                                     (SETQ BBLIST '(0 0 0 -1]
                                    (T (COND
                                          ((IGREATERP BBOX MAXBBOX)
                                           (SETQ MAXBBOX BBOX)))
                                       (COND
                                          ((ILESSP BBOX MINBBOX)
                                           (SETQ MINBBOX BBOX)))
                                       (COND
                                          ((IGREATERP BBOY MAXBBOY)
                                           (SETQ MAXBBOY BBOY)))
                                       (COND
                                          ((ILESSP BBOY MINBBOY)
                                           (SETQ MINBBOY BBOY)))
                                       [COND
                                          ((IGREATERP (IPLUS BBOX BBDX)
                                                      MAXSUMBBOXBBDX)
                                           (SETQ MAXSUMBBOXBBDX (IPLUS BBOX BBDX]
                                       (COND
                                          ((IGREATERP (IPLUS BBOY BBDY)
                                                      MAXSUMBBOYBBDY)
                                           (SETQ MAXSUMBBOYBBDY (IPLUS BBOY BBDY]
                                                             ; \GETFBB returns the fbbdx fbbdy fbbox fbboy of an acfont
          (RETURN (create FONTBOUNDINGBOX
                          FBBBDX _ (IDIFFERENCE MAXSUMBBOXBBDX MINBBOX)
                          FBBBDY _ (IDIFFERENCE MAXSUMBBOYBBDY MINBBOY)
                          FBBBOX _ MINBBOX
                          FBBBOY _ MINBBOY])
```

(\ACCHARPOSLIST

```
  [LAMBDA (FILE STARTCHAR ENDCHAR)                           (* jds "10-NOV-83 20:19")
                                                             ; \ACCHARPOSLIST returns the word position of the raster for
                                                             ; the nth character of the file
    [SETFILEPTR FILE (IPLUS 48 (ITIMES 16 (ADD1 (IDIFFERENCE ENDCHAR STARTCHAR]
```

```
(bind HIWORD LOWORD [DIRECTORYSTART _ (IPLUS 48 (ITIMES 16 (ADD1 (IDIFFERENCE ENDCHAR STARTCHAR]
   first (SETFILEPTR FILE DIRECTORYSTART) for X from STARTCHAR to ENDCHAR
   collect (SETQ HIWORD (\WIN FILE))
          (SETQ LOWORD (\WIN FILE))                          ; If the position of the acchar is given as -1,-1 then the raster
                                                             ; does not exist so return nil
          (COND
             ((AND (IEQP HIWORD 65535)
                   (IEQP LOWORD 65535))
              NIL)
             (T (IPLUS (LLSH HIWORD 17)
                       (LLSH LOWORD 1)
                      DIRECTORYSTART])
```

(\ACROTATECHAR

```
[LAMBDA (BITMAP)                                             ; Edited 28-Jul-87 18:49 by Snow

;; (prog (new.bitmap (width (|fetch| (bitmap bitmapwidth) |of| bitmap)) (height (|fetch| (bitmap bitmapheight) |of| bitmap))) (setq new.bitmap
;; (bitmapcreate height width)) (|for| y |from| 0 |to| (sub1 height) |do| (|for| x |from| 0 |to| (sub1 width) |bind| (y1 _ (idifference (sub1 height) y)) |do|
;; (bitmapbit new.bitmap y1 x (bitmapbit bitmap x y)))) (return new.bitmap))

(ROTATE-BITMAP-LEFT BITMAP])
```

(\READFONTWDFILE

```
[LAMBDA (FILE FD WIDTHS SCALE)                               (* jds " 2-Jan-86 12:34")

;; Widths array is fully allocated, with zeroes for characters with no information.  An array is not allocated for fixed WidthsY.  DEVICE is PRESS or
;; INTERPRESS

(DECLARE (GLOBALVARS FONTWIDTHSFILES))                       (* (RESETLST (* ; "RESETLST to make sure the fontfiles get
                                                             closed") (PROG (FIXEDFLAGS FIRSTCHAR LASTCHAR TEM
WIDTHSY) (SETFILEPTR FILE (LLSH (\FIXPIN FILE) 1)) (* ; "Locate the segment") (replace (FONTDESCRIPTOR FBBOX) of FD with
                                                             (SIGNED (\WIN FILE) BITSPERWORD))
                                                             (replace \SFDescent of FD with (IMINUS
                                                             (SIGNED (\WIN FILE) BITSPERWORD)))
                                                             (* ; "Descent is -FBBOY") (replace
                                                             (FONTDESCRIPTOR FBBDX) of FD with
                                                             (SIGNED (\WIN FILE) BITSPERWORD))
                                                             (replace \SFHeight of FD with (SIGNED
                                                             (\WIN FILE) BITSPERWORD)) (* ;
                                                             "Height is FBBDY") (replace \SFWidths of FD with WIDTHS)
                                                             (SETQ FIRSTCHAR (fetch FIRSTCHAR of FD))
                                                             (* ; "First and last 'real' characters in the font")
                                                             (SETQ LASTCHAR (fetch LASTCHAR of FD))
                                                             (COND (SCALE (* ; "Dimensions are relative, must be scaled")
                                                             (replace (FONTDESCRIPTOR FBBOX) of FD with
                                                             (IQUOTIENT (ITIMES (fetch (FONTDESCRIPTOR FBBOX) of
                                                             FD) SCALE) 1000)) (replace \SFDescent of FD with
                                                             (IQUOTIENT (ITIMES (fetch \SFDescent of FD) SCALE) 1000))
                                                             (replace (FONTDESCRIPTOR FBBDX) of FD with
                                                             (IQUOTIENT (ITIMES (fetch (FONTDESCRIPTOR FBBDX) of
                                                             FD) SCALE) 1000)) (replace \SFHeight of FD with
                                                             (IQUOTIENT (ITIMES (fetch \SFHeight of FD) SCALE) 1000))))
                                                             (replace \SFAscent of FD with (IDIFFERENCE
                                                             (fetch \SFHeight of FD) (fetch \SFDescent of FD)))
                                                             (SETQ FIXEDFLAGS (LRSH (\BIN FILE) 6))
                                                             (* ; "The fixed flags") (\BIN FILE) (* ;
                                                             "Skip the spares") (COND ((EQ 2
                                                             (LOGAND FIXEDFLAGS 2)) (SETQ TEM
                                                             (\WIN FILE)) (* ; "The fixed width for this font")
                                                             (COND ((AND SCALE (NOT (ZEROP TEM)))
                                                             (SETQ TEM (IQUOTIENT (ITIMES TEM SCALE) 1000))))
                                                             (for I from FIRSTCHAR to LASTCHAR do
                                                             (SETA WIDTHS I TEM))) (T (AIN WIDTHS FIRSTCHAR
                                                             (ADD1 (IDIFFERENCE LASTCHAR FIRSTCHAR)) FILE)
                                                             (for I from FIRSTCHAR to LASTCHAR when
                                                             (EQ noInfoCode (ELT WIDTHS I)) do
                                                             (SETA WIDTHS I 0)) (COND (SCALE
                                                             (for I from FIRSTCHAR to LASTCHAR do
                                                             (SETA WIDTHS I (IQUOTIENT (ITIMES
                                                             (ELT WIDTHS I) SCALE) 1000)))))))
                                                             (COND ((EQ 1 (LOGAND FIXEDFLAGS 1))
                                                             (SETQ WIDTHSY (\WIN FILE)) (* ;
                                                             "The fixed width-Y for this font;  the width-Y field is a single
                                                             integer in the FD") (replace \SFWidthsY of FD with
                                                             (COND ((AND SCALE (NOT (ZEROP WIDTHSY)))
                                                             (IQUOTIENT (ITIMES WIDTHSY SCALE) 1000))
                                                             (T WIDTHSY)))) (T (replace \SFWidthsY of FD with
                                                             (SETQ WIDTHSY (ARRAY (ADD1 \MAXCHAR)
                                                             (QUOTE SMALLPOSP) 0 0))) (AIN WIDTHSY FIRSTCHAR
                                                             (ADD1 (IDIFFERENCE LASTCHAR FIRSTCHAR)) FILE)
                                                             (for I from FIRSTCHAR to LASTCHAR when
                                                             (EQ noInfoCode (ELT WIDTHSY I)) do
                                                             (SETA WIDTHSY I 0)) (COND (SCALE
                                                             (for I from FIRSTCHAR to LASTCHAR do
                                                             (SETA WIDTHSY I (IQUOTIENT
                                                             (ITIMES (ELT WIDTHSY I) SCALE) 1000)))))))))))

(HELP])
```

(\FACECODE

```
  [LAMBDA (FACE)                                                    (* rmk%: "27-FEB-81 12:16")
    (IPLUS (SELECTQ (fetch (FONTFACE EXPANSION) of FACE)
                (REGULAR 0)
                (COMPRESSED 6)
                (EXPANDED 12)
                (SHOULDNT))
           (SELECTQ (fetch (FONTFACE WEIGHT) of FACE)
                (MEDIUM 0)
                (BOLD 2)
                (LIGHT 4)
                (SHOULDNT))
           (SELECTQ (fetch (FONTFACE SLOPE) of FACE)
                (REGULAR 0)
                (ITALIC 1)
                (SHOULDNT])
```

(\FAMILYCODE

```
  [LAMBDA (FAMILY WSTRM)                                            (* rmk%: "11-Sep-84 10:54")
    ;; Returns the family CODE for FAMILY in a standard widths file, leaving the file positioned at the beginning of the next file entry.  Returns NIL if
    ;; FAMILY not found.  If FAMILY is T, returns the code for the first family in the index.
    (SETFILEPTR WSTRM 0)
    (bind TYPE CODE LENGTH (NCHARS _ (NCHARS FAMILY))
          (NEXT _ 0) do (SETFILEPTR WSTRM NEXT)
                        (SETQ TYPE (\BIN WSTRM))
                        (SETQ LENGTH (\BIN WSTRM))
                        (add NEXT (LLSH (IPLUS LENGTH (LLSH (LOGAND TYPE 15)
                                                            8))
                                        1))
                        (SELECTQ (LRSH TYPE 4)
                             (1 (SETQ CODE (\WIN WSTRM))
                                (COND
                                   ([OR (EQ FAMILY T)
                                        (AND (EQ NCHARS (\BIN WSTRM))
                                             (for I from 1 to NCHARS always (EQ (\BIN WSTRM)
                                                                                (NTHCHARCODE FAMILY I]
                                    (SETFILEPTR WSTRM NEXT)       ; Move file to next entry
                                    (RETURN CODE))))
                             (0 (RETURN NIL))
                             NIL])
```

(\FINDFONT

```
  [LAMBDA (FD WSTRM PRESSMICASIZE NSMICASIZE DONTCHECK)             ; Edited  2-Apr-87 14:39 by bvm:
    ;; Finds the widths information for the specified FAMILY, FACECODE, MSIZE, and ROTATION.  The FIRSTCHAR and LASTCHAR of the font are
    ;; filled in, since we have to read past those to check the size.  If successful, returns the size found in the widths file, with zero indicating that
    ;; dimensions in the widths file are relative, leaving the file pointing just after the Rotation word of the font.  --- If DONTCHECK, then assumes that
    ;; this file contains exactly the right face and family, without checking --- Returns NIL if the font is not found
                                                                    (* (bind TYPE LENGTH SIZE FAMILYCODE
                                                                    (ROTATION _ (fetch ROTATION of FD))
                                                                    (FACECODE _ (\FACECODE (fetch FONTFACE of FD)))
                                                                    (NEXT _ 0) (FUZZ _ (PROG1 0.02
                                                                    (* ; "percentile difference acceptable as the same font size")))
                                                                    first (OR (SETQ FAMILYCODE (\FAMILYCODE
                                                                    (OR DONTCHECK (fetch FONTFAMILY of FD)) WSTRM))
                                                                    (RETURN NIL)) do (SETQ TYPE
                                                                    (\BIN WSTRM)) (SETQ LENGTH
                                                                    (\BIN WSTRM)) (add NEXT (LLSH
                                                                    (IPLUS LENGTH (LLSH (LOGAND TYPE 15) 8)) 1))
                                                                    (SELECTQ (LRSH TYPE 4) (4 (COND
                                                                    ((OR (AND (EQ FAMILYCODE (\BIN WSTRM))
                                                                    (EQ FACECODE (\BIN WSTRM))) DONTCHECK)
                                                                    (* ; "This is the right family/face (DONTCHECK must come last,
                                                                    so the file reads get done.)") (replace FIRSTCHAR of FD with
                                                                    (\BIN WSTRM)) (replace LASTCHAR of FD with
                                                                    (\BIN WSTRM)) (COND ((AND (OR
                                                                    (ZEROP (SETQ SIZE (\WIN WSTRM)))
                                                                    (LESSP (ABS (FQUOTIENT (IDIFFERENCE
                                                                    (OR PRESSMICASIZE NSMICASIZE) SIZE) PRESSMICASIZE)
) FUZZ)) (EQ ROTATION (\WIN WSTRM))) (replace \SFFACECODE of FD with FACECODE) (RETURN SIZE))))))) (0 (RETURN NIL)) NIL)
                                                                    (SETFILEPTR WSTRM NEXT)))
    (HELP])

)

(RPAQ? INTERPRESSFONTDIRECTORIES '("{Erinyes}<Lyric>Fonts>"))

(DECLARE%: EVAL@COMPILE

(PUTPROPS \POSITIONFONTFILE MACRO
          ((WSTRM NSMICASIZE FIRSTCHAR LASTCHAR FAMILY FACECODE) (* gbn "25-Jul-85 02:15")
                                                                    ; sets FIRSTCHAR LASTCHAR, and positions the file correctly
           ;; Finds the widths information for the specified FAMILY, FACECODE, MSIZE, and ROTATION.  FIRSTCHAR and LASTCHAR are
```

```
            ;; passed in since we have to read past those to check the size.  If successful, returns the size found in the widths file, with zero indicating
            ;; that dimensions in the widths file are relative, leaving the file pointing just after the Rotation word of the font.  --- --- Returns NIL if the
            ;; font is not found
            (bind TYPE LENGTH SIZE FAMCODE FILEFAM FILEFACE (NEXT _ 0)
               first (OR (SETQ FAMCODE (\FAMILYCODE (OR FAMILY T)
                                                    WSTRM))
                         (RETURN NIL))
               do (SETQ TYPE (\BIN WSTRM))
                  (SETQ LENGTH (\BIN WSTRM))
                  (add NEXT (LLSH (IPLUS LENGTH (LLSH (LOGAND TYPE 15)
                                                      8))
                                  1))
                  (SELECTQ (LRSH TYPE 4)
                       (4 (SETQ FILEFAM (\BIN WSTRM))
                          (SETQ FILEFACE (\BIN WSTRM))            ; This is the right family/face
                          [COND
                             ((OR (EQ FAMILY T)
                                  (EQ FAMILY NIL)
                                  (AND (IEQP FILEFAM FAMCODE)
                                       (IEQP FILEFACE FACECODE)))
                              (SETQ FIRSTCHAR (\BIN WSTRM))
                              (SETQ LASTCHAR (\BIN WSTRM))
                              (COND
                                 ((AND (OR (ZEROP (SETQ SIZE (\WIN WSTRM)))
                                           (LESSP (ABS (FQUOTIENT (IDIFFERENCE NSMICASIZE SIZE)
                                                                  NSMICASIZE))
                                                  0.02))
                                       (ZEROP (\WIN WSTRM)))
                                  (RETURN SIZE])
                       (0 (RETURN NIL))
                       NIL)
                  (SETFILEPTR WSTRM NEXT))))
)

(PUTPROPS AFONT COPYRIGHT ("Venue & Xerox Corporation" 1984 1985 1986 1987 1990))
```

FUNCTION INDEX

RECORD INDEX

MACRO INDEX

VARIABLE INDEX

CONSTANT INDEX

FILE-ENVIRONMENT INDEX