

File created: 15-Apr-2026 09:04:48 {WMEDLEY}<sources>ACFONT.;11

edit by: rmk

changes to: (VARS ACFONTCOMS)

previous date: 13-Apr-2026 09:00:05 {WMEDLEY}<sources>ACFONT.;10

Read Table: INTERLISP

Package: INTERLISP

Format: UTF-8

(RPAQQ ACFONTCOMS

;; AC and STRIKE font file support.

(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS BOUNDINGBOX FONTBOUNDINGBOX))
(FNS ACFONT.FILEP ACFONT.GETCHARSET \READACFONTBOXES \READACFONTFILE \ACCHARIMAGELIST
\ACCHARWIDTHHLIST \GETFBB \ACCHARPOSLIST \ACROTATECHAR \FACECODE \FAMILYCODE)
(PROP FILETYPE ACFONT)
[APPENDVARS (DISPLAYCHARSETFNS '(AC ACFONT.FILEP ACFONT.GETCHARSET)
(COMS ; STRIKE format files
(FNS STRIKEFONT.FILEP STRIKEFONT.GETCHARSET WRITESTRIKEFONTFILE STRIKECSINFO)
(APPENDVARS (DISPLAYCHARSETFNS '(STRIKE STRIKEFONT.FILEP STRIKEFONT.GETCHARSET))

;; AC and STRIKE font file support.

(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 (\* Height 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))
)

(DEFINEQ

(ACFONT.FILEP
[LAMBDA (FILE)
(RESETLST

; Edited 15-May-2025 17:48 by rmk

(CL:UNLESS (OPENP FILE 'INPUT)
[RESETSAVE (SETQ FILE (OPENSTREAM FILE 'INPUT 'OLD))
'(PROGN (CLOSEF? OLDVALUE))

;; This is the length of a standard index header. Other files could also have this value, but it's a pretty good discriminator

;; Skip to byte 25; do it with BINS so works for non-randaccessp devices. This skips the standard name header, then look for type 3 in the
;; following header

(CL:WHEN (EQ (\WIN FILE)
(LOGOR (LLSH 16 8)
12))
(FRPTQ 22 (\BIN FILE)) ; (SETFILEPTR STRM 25)
(EQ 3 (LRSH (\BIN FILE)
4))))))

(ACFONT.GETCHARSET
[LAMBDA (STRM CHARSET FONT)

; Edited 28-Mar-2026 23:02 by rmk
; Edited 27-Mar-2026 07:59 by rmk
; Edited 14-Jul-2025 19:50 by rmk
; Edited 17-May-2025 10:15 by rmk

;; STRM must be good for this CHARSET. This defaults the padding arguments of \READACFONTFILE

(READACFONTFILE STRM])

(\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. ; Move to the start of AC file's width info. ; Now collect the 4 bounding box values into a list

(SETFILEPTR FILE 48) (for X from STARTCHAR to ENDCHAR collect

(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 PAD.LEFT DONT.PAD.RIGHT)

; Edited 14-Jul-2025 19:49 by rmk ; Edited 8-Jul-2025 22:04 by rmk ; Edited 9-Jun-2025 14:17 by rmk ; Edited 16-May-2025 17:44 by rmk ; Edited 1-Sep-87 10:04 by Snow

(RESETLST (PROG [FBBLIST STARTCHAR ENDCHAR CHARWIDTHLIST CHARIMAGewidthLIST OFFSETS WIDTHS IMAGEWIDTHS FONTDESC FBBITMAP CHARBITMAP STARTWORDLIST BBOXLIST DUMMYCHAROFFSET DUMMYWIDTH (CSINFO (create CHARSETINFO

IMAGEWIDTHS ← ( \CREATECSINFOELEMENT ) LEFTKERN ← ( \CREATEKERNELEMENT ]

(CL:UNLESS (GETSTREAM STRM 'INPUT T) [RESETSAVE (SETQ STRM (OPENSTREAM STRM 'INPUT 'OLD)) '(PROGN (CLOSEF? OLDVALUE))

[COND ((AND (GETSTREAM STRM 'INPUT T) (RANDACCESSP STRM)) ; Presumably open from \READDISPLAYFONTFILE (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 0)

(CL:UNLESS (ACFONT.FILEP STRM) (ERROR "Not an AC font file" STRM))

(SETFILEPTR STRM 28) ; Starting at 28 skips the family and face bytes.

(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
            (SETQ DUMMYWIDTH 6)))
        ; Otherwise, make the dummy 6 wide.
      )))
  (COND
    ((NULL (REMOVE 0 CHARIMAGEWIDTHLIST))
      (ERROR "No raster mages" NIL)
      (RETURN)))
  (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 CSINFO 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))

  ;; 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)
            (fetch (FONTBOUNDINGBOX FBBBDY)
              of FBBLIST)
            (fetch (FONTBOUNDINGBOX FBBBOY)
              of FBBLIST)
            (fetch (FONTBOUNDINGBOX FBBBOY)
              of FBBLIST)
            (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)
                                                                 (CADDR 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))))))

```

[LAMBDA (BOXLIST)

(\* jds "15-Jun-85 11:37")

:: Returns a list of the ESCAPEMENTS (ie how far to move after printing 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  
FBBDX  
)  
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 fbbdy 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])

```

```

(\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])

)

(PUTPROPS ACFONT FILETYPE CL:COMPILE-FILE)

(APPENDTOVAR DISPLAYCHARSETFNS '(AC ACFONT.FILEP ACFONT.GETCHARSET))

:: STRIKE format files

(DEFINEQ

(STRIKEFONT.FILEP  
[LAMBDA (FILE)

; Edited 15-May-2025 17:47 by rmk

:: If high bit of type is on, then must be strike. If 2nd bit is on, must be strike-index, and we punt. We don't care about the 3rd bit

:: first word has high bits (onebit index fixed). Onebit means 'new-style font', index is 0 for simple strike, 1 for index, and fixed is if all chars have  
:: max width. Lisp doesn't care about 'fixed'

(RESETLST  
(CL:UNLESS (OPENP FILE 'INPUT)  
[RETSAVE (SETQ FILE (OPENSTREAM FILE 'INPUT 'OLD))  
'(PROGN (CLOSEF? OLDVALUE))  
(CL:WHEN [MEMB (\WIN FILE)  
(CONSTANT (LIST (LLSH 1 15)  
(LOGOR (LLSH 1 15)  
(LLSH 1 13]  
T)))]

(STRIKEFONT.GETCHARSET  
[LAMBDA (STRM)

; Edited 3-Aug-2025 22:27 by rmk  
; Edited 1-Aug-2025 23:50 by rmk  
; Edited 14-Jul-2025 19:52 by rmk  
; Edited 9-Jun-2025 14:22 by rmk  
; Edited 12-Jul-2022 09:19 by rmk  
; Edited 4-Dec-92 12:11 by jds

:: STRM has already been determined to be a vanilla strike-format file holding only the desired charset.

; returns a charsetinfo

(RESETLST  
(CL:UNLESS (\GETSTREAM STRM 'INPUT T)

```

[RESETSAVE (SETQ STRM (OPENSTREAM STRM 'INPUT 'OLD))
  ('PROGN (CLOSEF? OLDVALUE))
(SETFILEPTR STRM 0)
(CL:UNLESS (STRIKEFONT.FILEP STRM)
  (ERROR "Not a STRIKE font file" STRM))
(CL:UNLESS (EQ 2 (GETFILEPTR STRM))
  (SETFILEPTR STRM 2))
(LET (CSINFO NUMBCODES RW BITMAP OFFSETS FIRSTCHAR LASTCHAR HEIGHT WIDTHS)
  (SETQ CSINFO (create CHARSETINFO))
  (SETQ FIRSTCHAR (\WIN STRM)) ; minimum ascii code
  (SETQ LASTCHAR (\WIN STRM)) ; maximum ascii code
  (\WIN STRM) ; MaxWidth which isn't used by anyone.
  (\WIN STRM) ; number of words in this StrikeBody
  (replace (CHARSETINFO CHARSETASCENT) of CSINFO with (\WIN STRM))
  ; ascent in scan lines (=FBBdy+FBBoy)
  (replace (CHARSETINFO CHARSETDESCENT) of CSINFO with (\WIN STRM))
  ; descent in scan-lines (=FBBoy)
  (\WIN STRM) ; offset in bits (<0 for kerning, else 0, =FBBx)
  (SETQ RW (\WIN STRM)) ; raster width of bitmap
  ; height of bitmap

;; JDS 12/4/92: Apparently, these fields can be signed values, if all chars, e.g., ride above the base line.
(SETQ HEIGHT (IPLUS (SIGNED (fetch (CHARSETINFO CHARSETASCENT) of CSINFO)
  16)
  (SIGNED (fetch (CHARSETINFO CHARSETDESCENT) of CSINFO)
  16)))
(SETQ BITMAP (BITMAPCREATE (UNFOLD RW BITSPERWORD)
  HEIGHT))
(\BINS STRM (fetch BITMAPBASE of BITMAP)
  0
  (UNFOLD (ITIMES RW HEIGHT)
  BYTESPERWORD)) ; read bits into bitmap
(replace (CHARSETINFO CHARSETBITMAP) of CSINFO with BITMAP)
(SETQ NUMBCODES (IDIFFERENCE (ADD1 LASTCHAR)
  FIRSTCHAR))
(SETQ OFFSETS (fetch (CHARSETINFO OFFSETS) of CSINFO))

;; Initialize the offsets to 0, all but FIRSTCHAR to be replaced with the slug offset
(for I from 0 to (IPLUS \MAXTHINCHAR 2) do (\FSETOFFSET OFFSETS I 0))
(for I from FIRSTCHAR as J from 1 to NUMBCODES do
  ; J starts at 1 because we know that the offset of J=0 is 0 ?
  (\FSETOFFSET OFFSETS I (\WIN STRM)))
(for I (SLUGOFFSET ← (\WIN STRM)) from 0 to \MAXTHINCHAR when (EQ 0 (\FGETOFFSET OFFSETS
  I))
  unless (EQ I FIRSTCHAR) do (\FSETOFFSET OFFSETS I SLUGOFFSET) finally (\FSETOFFSET
  OFFSETS
  SLUGCHARINDEX
  SLUGOFFSET
  )

;; There's one more so that \FONTRESETCHARWIDTHS can get the
;; slug width, otherwise not necessary
(\FSETOFFSET
OFFSETS
(ADD1
  SLUGCHARINDEX
  )
(\WIN STRM)))

;; Initialize the widths to 0
(SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
(for I from 0 to (IPLUS \MAXTHINCHAR 2) do (\FSETWIDTH WIDTHS I 0))
(\FONTRESETCHARWIDTHS CSINFO 0 SLUGCHARINDEX)
(replace (CHARSETINFO IMAGEWIDTHS) of CSINFO with (fetch (CHARSETINFO WIDTHS) of CSINFO))
CSINFO)))

```

(WRITESTRIKEFONTFILE

[LAMBDA (FONT CHARSET FILE)

; Edited 30-Aug-2025 23:21 by rmk
; Edited 28-Aug-2025 15:09 by rmk
; Edited 24-Aug-2025 11:39 by rmk
; Edited 3-Aug-2025 22:33 by rmk
; Edited 22-May-2025 09:53 by rmk
; Edited 1-Feb-2025 12:27 by mth
; Edited 12-Jul-2022 14:36 by rmk
(\* kbr%: "21-Oct-85 15:08")
; Write strike FILE using info in FONT.

```
(CL:UNLESS (FONTP FONT)
  (LISPERROR "ILLEGAL ARG" FONT))
(CL:UNLESS CHARSET (SETQ CHARSET 0))
(CL:UNLESS (AND (IGEQ CHARSET 0)
  (ILEQ CHARSET \MAXCHARSET))
  (LISPERROR "ILLEGAL ARG" CHARSET))
(LET (STREAM CSINFO FIRSTCHAR LASTCHAR WIDTHS MAXWIDTH LENGTH RASTERWIDTH SLUGOFFSET OFFSETS)
  (SETQ CSINFO (\INSURECHARSETINFO FONT CHARSET))
  (CL:UNLESS CSINFO (ERROR "Couldn't find charset " CHARSET))
  (SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
  (SETQ OFFSETS (fetch (CHARSETINFO OFFSETS) of CSINFO))
  (SETQ SLUGOFFSET (\FGETOFFSET OFFSETS SLUGCHARINDEX))
  ;; Find the first and last non-slug characters
  [SETQ FIRSTCHAR (for I from 0 to \MAXTHINCHAR thereis (NEQ SLUGOFFSET (\FGETOFFSET OFFSETS I]
  [SETQ LASTCHAR (for I from \MAXTHINCHAR to 0 by -1 thereis (NEQ SLUGOFFSET (\FGETOFFSET OFFSETS I]
  [SETQ STREAM (OPENSTREAM FILE 'OUTPUT 'NEW '(TYPE BINARY)
  (\WOUT STREAM 32768) ; STRIKE HEADER.
  (\WOUT STREAM FIRSTCHAR)
  (\WOUT STREAM LASTCHAR)
  (SETQ MAXWIDTH 0)
  [for I from 0 to SLUGCHARINDEX do (SETQ MAXWIDTH (IMAX MAXWIDTH (\FGETWIDTH WIDTHS I]
  (\WOUT STREAM MAXWIDTH) ; STRIKE BODY.
  ; Length.
  (SETQ RASTERWIDTH (fetch (BITMAP BITMAPRASTERWIDTH) of (fetch (CHARSETINFO CHARSETBITMAP) of CSINFO)))
  (SETQ LENGTH (IPLUS 8 (DIFFERENCE LASTCHAR FIRSTCHAR)
  (TIMES (fetch (FONTDESCRIPTOR \SFHeight) of FONT)
  RASTERWIDTH)))
  (\WOUT STREAM LENGTH) ; Ascent, Descent, Xoffset (no longer used) and Rasterwidth.
  (\WOUT STREAM (fetch (CHARSETINFO CHARSETASCENT) of CSINFO))
  (\WOUT STREAM (fetch (CHARSETINFO CHARSETDESCENT) of CSINFO))
  (\WOUT STREAM 0)
  (\WOUT STREAM RASTERWIDTH) ; Bitmap.
  [BOUITS STREAM (fetch (BITMAP BITMAPBASE) of (fetch (CHARSETINFO CHARSETBITMAP) of CSINFO))
  0
  (TIMES 2 RASTERWIDTH (IPLUS (fetch (CHARSETINFO CHARSETASCENT) of CSINFO)
  (fetch (CHARSETINFO CHARSETDESCENT) of CSINFO)
  ; Offsets.
  [for I (OFFSET ← 0) from FIRSTCHAR to LASTCHAR first (\WOUT STREAM OFFSET)
  ; Offset of the first char
  do (CL:UNLESS (EQ SLUGOFFSET (\FGETOFFSET OFFSETS I)) ; The slug isn't really here in the bitmap
  (ADD OFFSET (\FGETWIDTH WIDTHS I]))
  (\WOUT STREAM OFFSET)
  finally ; Offset for the after-slug, for width
  (\WOUT STREAM (IPLUS OFFSET (\FGETWIDTH WIDTHS SLUGCHARINDEX]
  (CLOSEF STREAM))
```

(STRIKECSINFO
[LAMBDA (CSINFO)

; Edited 27-Apr-89 13:39 by atm

:: Returns a STRIKE type font descriptor (EQ WIDTHS IMAGEWIDTHS), cause we know how to write those guys out (they read quicker but
:: display slower). If (EQ WIDTHS IMAGEWIDTHS), just return original.

```
(PROG (WIDTHS OFFSETS IMWIDTHS OLDBM BMWIDTH BMHEIGHT NEWBM NEWOFFSET NEWWIDTH
  OLDOFFSET DUMMYOFFSET NEWOFFSETS)
  (SETQ WIDTHS (fetch (CHARSETINFO WIDTHS) of CSINFO))
  (SETQ IMWIDTHS (fetch (CHARSETINFO IMAGEWIDTHS) of CSINFO))
  (if (EQ WIDTHS IMWIDTHS)
    then (RETURN CSINFO))
  (SETQ OFFSETS (fetch (CHARSETINFO OFFSETS) of CSINFO))
  (SETQ OLDBM (fetch (CHARSETINFO CHARSETBITMAP) of CSINFO))
  (SETQ DUMMYOFFSET (\FGETOFFSET OFFSETS 256))
  (SETQ BMHEIGHT (BITMAPHEIGHT OLDBM))
  [SETQ BMWIDTH (for I from 0 to \MAXTHINCHAR sum (if (IEQP DUMMYOFFSET (\FGETOFFSET OFFSETS
  I))
  then 0
  else (IMAX (\FGETIMAGEWIDTH IMWIDTHS I]
  (\FGETWIDTH WIDTHS I]
  ;
```

::
:: Initialize new offsets vector

::

```
(SETQ NEWOFFSETS (\CREATECSINFOELEMENT))
(for I from 0 to (IPLUS \MAXTHINCHAR 2) do (\FSETOFFSET NEWOFFSETS I 0))
(\FSETOFFSET NEWOFFSETS (ADD1 \MAXTHINCHAR)
  BMWIDTH)
```

::

:: Adjust bitmap with so width = imagewidth, fill offsets

::

```
(SETQ NEWBM (BITMAPCREATE BMWIDTH BMHEIGHT 1))
(SETQ NEWOFFSET 0)
[for I from 0 to 255 do (SETQ OLDOFFSET (\FGETOFFSET OFFSETS I))
  (if (IEQP DUMMYOFFSET OLDOFFSET)
    then (\FSETOFFSET NEWOFFSETS I BMWIDTH)
    else (\FSETOFFSET NEWOFFSETS I NEWOFFSET)
      (SETQ NEWWIDTH (IMAX (\FGETIMAGEWIDTH IMWIDTHS I)
        (\FGETWIDTH WIDTHS I)))
      (BITBLT OLDBM OLDOFFSET 0 NEWBM NEWOFFSET 0 (\FGETWIDTH
        IMWIDTHS I)
        BMHEIGHT
        'REPLACE)
      (SETQ NEWOFFSET (IPLUS NEWOFFSET NEWWIDTH])
```

::

:: Make new CSInfo record with IMAGEWIDTHS, WIDTHS the same

::

```
(SETQ WIDTHS (COPYALL WIDTHS))
[for I from 0 to \MAXTHINCHAR do (\FSETWIDTH WIDTHS I (IMAX (\FGETWIDTH WIDTHS I)
  (\FGETIMAGEWIDTH IMWIDTHS I])
(RETURN (create CHARSETINFO
  WIDTHS ← WIDTHS
  OFFSETS ← NEWOFFSETS
  IMAGEWIDTHS ← WIDTHS
  CHARSETBITMAP ← NEWBM
  YWIDTHS ← (fetch (CHARSETINFO YWIDTHS) of CSINFO)
  CHARSETASCENT ← (fetch (CHARSETINFO CHARSETASCENT) of CSINFO)
  CHARSETDESCENT ← (fetch (CHARSETINFO CHARSETDESCENT) of CSINFO])
```

)

```
(APPENDTOVAR DISPLAYCHARSETFNS '(STRIKE STRIKEFONT.FILEP STRIKEFONT.GETCHARSET))
```

---

FUNCTION INDEX

ACFONT.FILE .....1	WRITESTRIKEFONTFILE .....8	\FACECODE .....6
ACFONT.GETCHARSET .....1	\ACCHARIMAGELIST .....4	\FAMILYCODE .....7
STRIKECSINFO .....9	\ACCHARPOSLIST .....6	\GETFBB .....5
STRIKEFONT.FILE .....7	\ACCHARWIDTHLIST .....5	\READACFONTBOXES .....2
STRIKEFONT.GETCHARSET .....7	\ACROTATECHAR .....6	\READACFONTFILE .....2

---

RECORD INDEX

BOUNDINGBOX .....1	FONTBOUNDINGBOX .....1
--------------------	------------------------

---

PROPERTY INDEX

ACFONT .....7

---