

File created: 24-Sep-2023 14:29:56 {WMEDLEY}<lispusers>SHOWTIME.;2

edit by: rmk

changes to: (VARS SHOWTIMECOMS SHOWTIME.ICON SHOWTIME.MASK)  
(FNS INFORES SHOWTIME.READ.PRESS)

previous date: 10-Apr-89 18:56:29 {WMEDLEY}<lispusers>SHOWTIME.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;  
;; Copyright (c) 1986-1989 by Xerox Corporation.

(RPAQQ **SHOWTIMECOMS**  
(

;;; Mitch Gaarnat and (Mike?) Gocek wrote the original versions of these fns in 1985. They were later modified added to by T. Bigham in 1986 and 1987.  
;;; Ron Fischer at Xerox AI Systems made a quick pass to convert the file to run in Medley XAIE.

```
(FNS GET.SHOWTIME.MENU MAKEBRUSH MAKEBRUSH.HEADER&BITMAP INFORES READ.RES SHOWTIME
SHOWTIME.BUTTONEVENTFN SHOWTIME.GET.NAME SHOWTIME.ICONFN SHOWTIME.LOAD.BITMAP SHOWTIME.LOAD.BRUSH
SHOWTIME.LOAD.DIF.FILE SHOWTIME.LOAD.RES.FILE SHOWTIME.MAKE.RES SHOWTIME.MAKE.RES.HEADER
SHOWTIME.MAKE.RES.TAIL SHOWTIME.READ.BRUSH SHOWTIME.READ.LISPBM SHOWTIME.READ.PRESS
SHOWTIME.READ.RES SHOWTIME.RES.CHECK&MESSAGE SHOWTIME.RESHAPE.WINDOW SHOWTIME.SAVE.BITMAP
SHOWTIME.SAVE.LISPBM SHOWTIME.SCALE.BITMAP SHOWTIME.ADD.FORMAT SHOWTIME.SETUP.WINDOWPROPS
SHOWTIME.SHOW.BITMAP SHOWTIME.WRITEBM)
[VARS SHOWTIME.ICON SHOWTIME.MASK (SHOWTIME.LOAD.SUBITEMS)
(SHOWTIME.SAVE.SUBITEMS)
(SHOWTIME.MENU)
(SHOWTIMETITLEREGION '(7 7 56 29))
(SHOWTIME.DEFAULT.FORMAT 'LISP)
(BackgroundMenu)
(SHOWTIME.FORMAT.FNS '(SHOWTIME.FORMAT.FNS (RES READ.RES SHOWTIME.MAKE.RES)
(LISP SHOWTIME.READ.LISPBM SHOWTIME.SAVE.LISPBM)
(DIF SHOWTIME.LOAD.DIF.FILE NIL)
(BRUSH SHOWTIME.LOAD.BRUSH MAKEBRUSH)
(PRESS READPRESS PRESSBITMAP)
(APPENDVARS (BackgroundMenuCommands (Showtime '(SHOWTIME)
"Opens a showtime window for use.)))
(FILES BITMAPFNS SCALEBITMAP READBRUSH)
(P (SHOWTIME.ADD.FORMAT)))
```

;;; Mitch Gaarnat and (Mike?) Gocek wrote the original versions of these fns in 1985. They were later modified added to by T. Bigham in 1986 and 1987.  
;;; Ron Fischer at Xerox AI Systems made a quick pass to convert the file to run in Medley XAIE.

(DEFINEQ

(**GET.SHOWTIME.MENU**

```
[LAMBDA NIL ; Edited 13-May-88 15:19 by raf
(DECLARE (GLOBALVARS SHOWTIME.MENU)
(OR SHOWTIME.MENU (SETQ SHOWTIME.MENU (create MENU
TITLE _ "Viewer Options"
CENTERFLG _ T
MENUFONT _ (FONTCREATE 'TIMESROMAN 12)
ITEMS _ `((" Load Another Picture " 'LOAD NIL
(SUBITEMS ,@SHOWTIME.LOAD.SUBITEMS)
["Show a Bitmap" 'SHOW NIL (SUBITEMS
("Put bitmap in this
window" 'SHOW)
("Reshape window to fit
new Bitmap"
'RESHAPE&SHOW]
("Edit Bitmap" 'EDIT)
["Redisplay" 'REDISPLAY NIL
(SUBITEMS ("Redisplay bitmap in this window"
'REDISPLAY)
("Reshape window to fit this bitmap"
'RESHAPE&SHOW]
("Scale" 'SCALE)
("Save this Picture" 'SAVE NIL (SUBITEMS
,@
SHOWTIME.SAVE.SUBITEMS
]))
```

(**MAKEBRUSH**

```
[LAMBDA (FILESPEC BITMAP) ; Edited 13-May-88 16:00 by raf
(LET [(STREAM (GETSTREAM FILESPEC 'OUTPUT)
(MAKEBRUSH.HEADER&BITMAP BITMAP STREAM)
(CLOSEF STREAM)])
```

**(MAKEBRUSH.HEADER&BITMAP**

; Edited 13-May-88 16:01 by raf

```
[LAMBDA (BITMAP STREAM)
  (LET ((BASE (fetch BITMAPBASE of BITMAP))
        (HEIGHT (fetch BITMAPHEIGHT of BITMAP))
        (WIDTH (fetch BITMAPWIDTH of BITMAP))
        (RWIDTH (fetch BITMAPRASTERWIDTH of BITMAP)))
    (BOUT STREAM 1)
    (BOUT STREAM 0)
    (BOUT STREAM (LRSH (LOGAND WIDTH 65280)
                        8))
    (BOUT STREAM (LOGAND WIDTH 255))
    (BOUT STREAM (LRSH (LOGAND HEIGHT 65280)
                        8))
    (BOUT STREAM (LOGAND HEIGHT 255))
    (for I from 1 to 8 do (BOUT STREAM 0))
    (BOUT STREAM (LRSH (LOGAND RWIDTH 65280)
                        8))
    (BOUT STREAM (LOGAND RWIDTH 255))
    (for Y from 0 to (SUB1 HEIGHT) do (for X from 0 to (SUB1 RWIDTH) do (BOUT STREAM (\GETBASEBYTE BASE 0))
                                     (BOUT STREAM (\GETBASEBYTE BASE 1))
                                     (SETQ BASE (\ADDBASE BASE 1]))
```

**(INFORES**

; Edited 24-Sep-2023 14:28 by rmk

; Edited 13-May-88 16:01 by raf

```
[LAMBDA (FILE)
  (LET (STREAM PATTERN WIDTH HEIGHT HI.X LO.X HI.Y LO.Y REAL.X REAL.Y (Header
                                                                    Interpress/Xerox/2.1/RasterEncoding/1.0%
                                                                    ))
    ; Return the width, height, bits per pixel and address of the first
    ; data byte as a list.
    (SETQ STREAM (OPENSTREAM FILE 'INPUT))
    (if [EQ Header (PACK (for X from 1 to 40 collect (CHARACTER (\BIN STREAM]
                                                                ; bypass BEGIN 254/720000 DUP 2 MAKEVEEC
                                                                then
                                                                [until (EQUAL (NTH (REVERSE PATTERN)
                                                                (IDIFFERENCE (LENGTH PATTERN)
                                                                4))
                                                                '(181 15 162 161 27))
                                                                do (SETQ PATTERN (push PATTERN (\BIN STREAM]
                                                                (SETQ HI.X (\BIN STREAM))
                                                                (SETQ LO.X (\BIN STREAM))
                                                                (SETQ HI.Y (\BIN STREAM))
                                                                (SETQ LO.Y (\BIN STREAM))
                                                                (SETQ REAL.X (IDIFFERENCE (PLUS (LSH HI.X 8)
                                                                LO.X)
                                                                4000))
                                                                (SETQ REAL.Y (IDIFFERENCE (PLUS (LSH HI.Y 8)
                                                                LO.Y)
                                                                4000))
                                                                (LIST REAL.X REAL.Y STREAM)
                                                                else (CLOSEF STREAM)
                                                                NIL))
```

**(READ.RES**

; Edited 13-May-88 16:02 by raf

; If the file exists, check to see if it's RES format.

```
[LAMBDA (FILE)
  (LET (STREAM A B BITMAP BASE WORDS Attributes WIDTH HEIGHT)
    (if (SETQ FILE (FULLNAME FILE))
      then
        (if (SETQ Attributes (INFORES FILE))
          then (SETQ WIDTH (CAR Attributes))
               (SETQ HEIGHT (CADR Attributes))
               (SETQ STREAM (CADDR Attributes))
               (SETQ BITMAP (BITMAPCREATE WIDTH HEIGHT 1))
               (SETQ BASE (fetch (BITMAP BITMAPBASE) of BITMAP))
               ; RESINFO leaves the file open at byte 62.0 Image data begins
               ; at byte 95
               (for X from 63 to 94 do (\BIN STREAM))
               (for X from 1 to (IQUOTIENT (ITIMES WIDTH HEIGHT)
                                             16)
                 do (SETQ A (\BIN STREAM))
                    (SETQ B (\BIN STREAM))
                    (\PUTBASE BASE 0 (LOGOR (LLSH A 8)
                                             B))
                    (SETQ BASE (\ADDBASE BASE 1))
                    (ZEROP (LOGAND X 1023)))
               (CLOSEF STREAM)
               BITMAP
          else (printout PROMPTWINDOW T FILE "isn't an RES file"))
      else (printout PROMPTWINDOW T "Can't find " FILE)
      NIL))
```

**(SHOWTIME**

```

[LAMBDA (FILENAME FORMAT)
  (LET (BITMAP WIDTH HEIGHT WINDOW BITMAP.NAME)
    (if FILENAME
      then (SETQ FILENAME (\ADD.CONNECTED.DIR FILENAME))
            (if (BITMAPP (SETQ BITMAP (SHOWTIME.LOAD.BITMAP FILENAME FORMAT)))
              then (SETQ WIDTH (BITMAPWIDTH BITMAP))
                    (SETQ HEIGHT (BITMAPHEIGHT BITMAP))
                    (SETQ WINDOW (CREATEW (LIST 200 200 WIDTH HEIGHT)))
                    (WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION SHOWTIME.BUTTONEVENTFN))
                    (WINDOWPROP WINDOW 'ICONFN (FUNCTION SHOWTIME.ICONFN))
                    (CLEARW WINDOW)
                    (BITBLT BITMAP 0 0 WINDOW)
                    (SETQ BITMAP.NAME (SET (U-CASE (FILENAMEFIELD FILENAME 'NAME))
                                          BITMAP))
                  (SHOWTIME.SETUP.WINDOWPROPS WINDOW BITMAP FILENAME)
              else (PROMPTPRINT (CONCAT "Couldn't find " FILENAME)))
            else (SETQ WINDOW (CREATEW NIL "Empty Viewer"))
                  (WINDOWPROP WINDOW 'BUTTONEVENTFN (FUNCTION SHOWTIME.BUTTONEVENTFN))
                  (WINDOWPROP WINDOW 'ICONFN (FUNCTION SHOWTIME.ICONFN))
                  (WINDOWPROP WINDOW 'BITMAP.VALUE NIL)
                  (WINDOWPROP WINDOW 'BITMAP.NAME NIL))
    )
  )

```

(\* TBigham "7-Feb-87 18:05")

(SHOWTIME.BUTTONEVENTFN

; Edited 13-May-88 15:35 by raf

```

[LAMBDA (WINDOW)
  (DECLARE (GLOBALVARS PROMPTWINDOW))
  (LET (BITMAP SELECTION BITMAP.NAME ICON PERIOD.POS SWITCH)
    (SETQ SELECTION (MENU (GET.SHOWTIME.MENU)))
    (if (SETQ PERIOD.POS (STRPOS "." SELECTION))
      then [SETQ SWITCH (MKATOM (SUBSTRING SELECTION 1 (SUB1 PERIOD.POS))
                                else (SETQ SWITCH SELECTION))
          (SELECTQ SWITCH
            (LOAD (SHOWTIME.LOAD.BITMAP NIL (MKATOM (SUBSTRING SELECTION 6))
              WINDOW))
            (SHOW (SETQ BITMAP.NAME (SHOWTIME.GET.NAME "Bitmap Name (CR to end)"))
              (if BITMAP.NAME
                then (SETQ BITMAP (EVAL BITMAP.NAME))
                     (SHOWTIME.SHOW.BITMAP BITMAP BITMAP.NAME WINDOW SELECTION))
              (RESHAPE&SHOW (SETQ BITMAP (WINDOWPROP WINDOW 'BITMAP))
                (if (BITMAPP BITMAP)
                  then (SHOWTIME.SHOW.BITMAP BITMAP (WINDOWPROP WINDOW 'BITMAP.NAME)
                    WINDOW SELECTION))
                (RESHAPE (SHOWTIME.RESHAPE.WINDOW BITMAP WINDOW)
                  (CLEARW WINDOW)
                  (BITBLT BITMAP 0 0 WINDOW))
                (REDISPLAY (SETQ BITMAP (WINDOWPROP WINDOW 'BITMAP))
                  (CLEARW WINDOW)
                  (BITBLT BITMAP 0 0 WINDOW))
                (EDIT (if [BITMAPP (SETQ BITMAP (WINDOWPROP WINDOW 'BITMAP))
                  then (SETQ BITMAP (EDIT.BITMAP BITMAP))
                       (WINDOWPROP WINDOW 'BITMAP BITMAP)
                       (CLEARW WINDOW)
                       (BITBLT BITMAP 0 0 WINDOW)
                  else (printout PROMPTWINDOW T "No bitmap in Viewer to edit!"))
                (SAVE [if [BITMAPP (SETQ BITMAP (WINDOWPROP WINDOW 'BITMAP))
                  then (SHOWTIME.SAVE.BITMAP NIL BITMAP (MKATOM (SUBSTRING SELECTION 6))
                    (SCALE (SHOWTIME.SCALE.BITMAP (WINDOWPROP WINDOW 'BITMAP)
                      WINDOW))
                    NIL)
                (if (SETQ ICON (WINDOWPROP WINDOW 'ICONWINDOW))
                  then (ICONW.TITLE ICON BITMAP.NAME])
    )
  )

```

(SHOWTIME.GET.NAME

(\* TBigham "27-Mar-86 10:42")

```

[LAMBDA (PROMPTSTRING USE.EOL)
  (DECLARE (GLOBALVARS LASTMOUSEX LASTMOUSEY))
  (OR PROMPTSTRING (SETQ PROMPTSTRING "Name (CR to end) "))
  (LET* ((FONT (DEFAULTFONT))
         [WIDTH (IPLUS (STRINGWIDTH PROMPTSTRING FONT)
           (ITIMES 60 (CHARWIDTH (CHARCODE A)
             FONT))
         (PROMPTW (CREATEW [MAKEWITHINREGION (CREATEREGION LASTMOUSEX LASTMOUSEY (WIDTHIFWINDOW WIDTH)
             (HEIGHTIFWINDOW (FONTPROP FONT 'HEIGHT)
               NIL NIL T)))
         (if USE.EOL
           then (RESETLST
             (RESETSAVE (OPENW PROMPTW)
               (CLOSEW %, PROMPTW))
             (MKATOM (PROMPTFORWARD PROMPTSTRING NIL NIL PROMPTW NIL NIL (CHARCODE EOL))))
           else (RESETLST
             (RESETSAVE (OPENW PROMPTW)
               (CLOSEW %, PROMPTW))
             (MKATOM (PROMPTFORWARD PROMPTSTRING NIL NIL PROMPTW NIL))))
    )
  )

```

(SHOWTIME.ICONFN

(\* TBigham "7-Feb-87 18:10")

```

[LAMBDA (WINDOW OLDICON)

```



```

then (SETQ BITMAP (SHOWTIME.RES.CHECK&MESSAGE BITMAP))
      (SHOWTIME.MAKE.RES.HEADER STREAM BITMAP)
      (SHOWTIME.WRITEBM BITMAP STREAM)
      (SHOWTIME.MAKE.RES.TAIL STREAM)
      (SETFILEINFO (CLOSEP? STREAM)
        'TYPE 4428])

```

(SHOWTIME.MAKE.RES.HEADER

(\* TBigham "26-Jun-87 09:11")

```

[LAMBDA (STREAM BITMAP)
  (LET* ((WIDTH (BITMAPWIDTH BITMAP))
         (HEIGHT (BITMAPHEIGHT BITMAP))
         (HEADER 'Interpress/Xerox/2.1/RasterEncoding/1.0% )
         (OFFSETWIDTH (PLUS 4000 WIDTH))
         (OFFSETHEIGHT 400 HEIGHT)
         (CLEANX (LOGAND 65535 (PLUS 4000 WIDTH)))
         (CLEANY (LOGAND 65535 (PLUS 4000 HEIGHT)))
         (LO.X.ICS (LOGAND 255 CLEANX))
         (HI.X.ICS (RSH CLEANX 8))
         (LO.Y.ICS (LOGAND 255 CLEANY))
         (HI.Y.ICS (RSH CLEANY 8))
         BOUNDARY)
    (SETQ BOUNDARY (for NUMBER from 32 by 32 thereis (if (ILEQ WIDTH NUMBER)
                                                         then NUMBER
                                                         else NIL)))
    (for CHAR in (UNPACK HEADER) do (PRINTCCODE (CAR (CHCON CHAR))
                                                STREAM)) (* BEGIN |254/720000| DUP 2 MAKEVEC)
    (for BYTE
      in '(160 102 196 8 0 0 0 254 0 10 252 128 160 181 15 162 161 27) do (\BOUT STREAM BYTE))
      (* x dimension in TCS)
    (\BOUT STREAM HI.X.ICS)
    (\BOUT STREAM LO.X.ICS) (* y dimension in TCS)
    (\BOUT STREAM HI.Y.ICS)
    (\BOUT STREAM LO.Y.ICS) (* mask image -- null)
    (\BOUT STREAM '15)
    (\BOUT STREAM '160) (* x dimension in ICS same as y dimension in TCS)
    (\BOUT STREAM HI.Y.ICS)
    (\BOUT STREAM LO.Y.ICS) (* y dimension in ICS same as x dimension in TCS)
    (\BOUT STREAM HI.X.ICS)
    (\BOUT STREAM LO.X.ICS) (* samples per pixel)
    (\BOUT STREAM '15)
    (\BOUT STREAM '161) (* max sample value)
    (\BOUT STREAM '15)
    (\BOUT STREAM '161) (* samples interleaved)
    (\BOUT STREAM '15)
    (\BOUT STREAM '160) (* first transformation from ICS to TCS)
    (\BOUT STREAM '15) (* -90 rotate)
    (\BOUT STREAM '70) (* second transformation from ICS to TCS)
    (\BOUT STREAM '160)
    (\BOUT STREAM '163) (* short integer)
    (\BOUT STREAM '15)
    (\BOUT STREAM '160) (* sense of the transform)
    (\BOUT STREAM HI.Y.ICS)
    (\BOUT STREAM LO.Y.ICS) (* translate transform)
    (\BOUT STREAM '160)
    (\BOUT STREAM '162) (* CONCAT)
    (\BOUT STREAM '160)
    (\BOUT STREAM '165)
    (\BOUT STREAM '233)
    (LET (HI.NIBBLE MID.NIBBLE LO.NIBBLE PACKED.PIXEL.VECTOR.SIZE CLEANNUM)
      (SETQ PACKED.PIXEL.VECTOR.SIZE (PLUS 4 (IQUOTIENT (ITIMES BOUNDARY HEIGHT)
                                                         8)))
      (SETQ CLEANNUM (LOGAND 16777215 PACKED.PIXEL.VECTOR.SIZE))
      (SETQ HI.NIBBLE (RSH (LOGAND 16711680 CLEANNUM)
                          16))
      (SETQ MID.NIBBLE (RSH (LOGAND 65280 CLEANNUM)
                           8))
      (SETQ LO.NIBBLE (LOGAND 255 CLEANNUM))
      (\BOUT STREAM HI.NIBBLE)
      (\BOUT STREAM MID.NIBBLE)
      (\BOUT STREAM LO.NIBBLE)) (* bits per sample)
    (\BOUT STREAM '0)
    (\BOUT STREAM '1)
    (\BOUT STREAM HI.X.ICS)
    (\BOUT STREAM LO.X.ICS])

```

(SHOWTIME.MAKE.RES.TAIL

(\* TBigham " 4-Apr-86 06:38")

```

[LAMBDA (STREAM)
  (LET [(DATA '(161 194 15 160 15 161 15 160 15 163 197 5 88 101 114 111 120 197 10 71 114 97 121 76 105 110
               101 97 114 15 162 161 27 161 166 160 231 15 160 66 190 160 103])
    (for BYTE in DATA do (BOUT STREAM BYTE]))

```

(SHOWTIME.READ.BRUSH

(\* TBigham "30-Dec-86 12:00")

```

[LAMBDA (FILENAME)
  (CAR (READBRUSHFILE FILENAME))]

```

**(SHOWTIME.READ.LISPBM**

```
[LAMBDA (FILENAME)
  (LET [(STREAM (OPENSTREAM FILENAME 'INPUT)
        (if (STREAMP STREAM)
            then (PROG1 (READBM STREAM)
                        (CLOSEF STREAM))
            else NIL])
```

; Edited 10-Apr-89 18:51 by Briggs

**(SHOWTIME.READ.PRESS**

```
[LAMBDA (FILENAME)
  (READPRESS FILENAME)]
```

; Edited 24-Sep-2023 14:29 by rmk  
(\* TBigham "30-Dec-86 11:59")

**(SHOWTIME.READ.RES**

```
[LAMBDA (FILENAME)
  (DECLARE (GLOBALVARS WAITINGCURSOR))
  (LET (BITMAP)
    (RESETLST
      (RESETSAVE (CURSOR WAITINGCURSOR))
      (SETQ BITMAP (READ.RES FILENAME))))]
```

(\* TBigham "30-Dec-86 12:03")  
(\* load an RES image and makes it into a lisp bitmap)

**(SHOWTIME.RES.CHECK&MESSAGE**

```
[LAMBDA (BITMAP)
  (LET ((HEIGHT (fetch BITMAPHEIGHT of BITMAP))
        (WIDTH (fetch BITMAPWIDTH of BITMAP))
        BOUNDARY BM)
```

(\* TBigham " 1-Mar-87 19:31")

(\* if the bitmap doesn't fit on a 32 bit boundary, extend the bitmap to do so)

```
(if (ZEROP (REMAINDER WIDTH 32))
  then BITMAP
  else (SETQ BOUNDARY (for NUMBER from 32 by 32 thereis (if (ILEQ WIDTH NUMBER)
                                                             then NUMBER)))
      (SETQ BM (BITMAPCREATE BOUNDARY HEIGHT))
      (BITBLT BITMAP 0 0 BM)
      BM])
```

**(SHOWTIME.RESHAPE.WINDOW**

```
[LAMBDA (BITMAP WINDOW)
  (LET NIL (SHAPEW WINDOW (CREATEREGION (CAR (WINDOWPROP WINDOW 'REGION))
                                         (CADR (WINDOWPROP WINDOW 'REGION))
                                         (WIDTHIFWINDOW (BITMAPWIDTH BITMAP))
                                         (HEIGHTIFWINDOW (BITMAPHEIGHT BITMAP)
                                                           T]))]
```

(\* TBigham "10-Mar-86 15:51")

**(SHOWTIME.SAVE.BITMAP**

```
[LAMBDA (FILENAME BITMAP FORMAT)
  (DECLARE (GLOBALVARS SHOWTIME.DEFAULT.FORMAT PROMPTWINDOW SHOWTIME.FORMAT.FNS WAITINGCURSOR))
  (OR FILENAME (SETQ FILENAME (SHOWTIME.GET.NAME NIL T)))
  (OR FORMAT (SETQ FORMAT SHOWTIME.DEFAULT.FORMAT))
  (LET [(SAVEFN (CADDR (FASSOC FORMAT SHOWTIME.FORMAT.FNS)
                          (if (NULL FILENAME)
                              THEN (printout PROMPTWINDOW "Null filename.")
                              ELSEIF (NOT (BITMAPP BITMAP))
                              THEN (printout PROMPTWINDOW BITMAP " not a bitmap.")
                              ELSEIF (NULL SAVEFN)
                              THEN (printout PROMPTWINDOW "Can't write bitmaps out in " FORMAT " format.")
                              ELSE (RESETLST
                                    (RESETSAVE (CURSOR WAITINGCURSOR))
                                    (printout PROMPTWINDOW T "Saving " FILENAME "...")
                                    (EVAL (LIST SAVEFN (KWOTE FILENAME)
                                                (KWOTE BITMAP)))
                                    (printout PROMPTWINDOW "done!")))]
```

; Edited 13-May-88 15:56 by raf

**(SHOWTIME.SAVE.LISPBM**

```
[LAMBDA (FILENAME BITMAP)
  (LET (STREAM)
    (if [AND FILENAME (SETQ STREAM (OPENSTREAM FILENAME 'OUTPUT)]
        then (WRITEBM STREAM BITMAP)
        (CLOSEF STREAM])
```

; Edited 13-May-88 15:59 by raf

**(SHOWTIME.SCALE.BITMAP**

```
[LAMBDA (BITMAP WINDOW)
  (DECLARE (GLOBALVARS PROMPTWINDOW WAITINGCURSOR))
  (LET (XSCALE YSCALE SCALE)
    (if (BITMAPP BITMAP)
        then (OR (SETQ XSCALE (RNUMBER "X SCALE FACTOR" NIL NIL NIL T T))
```

(\* TBigham "31-Mar-86 15:42")

```

      (SETQ XSCALE 1))
    (OR (SETQ YSCALE (RNUMBER "Y SCALE FACTOR" NIL NIL NIL T T))
        (SETQ YSCALE 1))
    (SETQ SCALE (CONS (FLOAT XSCALE)
                      (FLOAT YSCALE)))
    (RESETLST
     (RESETSAVE (CURSOR WAITINGCURSOR))
     (SETQ BITMAP (SCALEBITMAP BITMAP SCALE)))
    (WINDOWPROP WINDOW 'BITMAP BITMAP)
    (SETTOPVAL (WINDOWPROP WINDOW 'BITMAP.NAME)
               BITMAP)
    (CLEARW WINDOW)
    (BITBLT BITMAP 0 0 WINDOW)
  else (printout PROMPTWINDOW T (CONCAT BITMAP " not a bitmap."))

```

**(SHOWTIME.ADD.FORMAT**

```

[LAMBDA (FORMAT READFN WRITEFN) (* TBigham "30-Dec-86 13:11")
  (DECLARE (GLOBALVARS SHOWTIME.FORMAT.FNS SHOWTIME.LOAD.SUBITEMS SHOWTIME.MENU SHOWTIME.SAVE.SUBITEMS))

```

(\* adds information to an assoc list which identifies read and write functions for different bitmap formats)

```

(LET NIL (if FORMAT
  then (PUTASSOC FORMAT (LIST READFN WRITEFN)
                    SHOWTIME.FORMAT.FNS)
  [SETQ SHOWTIME.LOAD.SUBITEMS (for EACH in (CDR SHOWTIME.FORMAT.FNS)
    collect (LIST (CONCAT (CAR EACH)
                          " Format")
                  (KWOTE (MKATOM (CONCAT "LOAD." (CAR EACH))
                              " Format"))
  [SETQ SHOWTIME.SAVE.SUBITEMS (for EACH in (CDR SHOWTIME.FORMAT.FNS)
    collect (LIST (CONCAT (CAR EACH)
                          " Format")
                  (KWOTE (MKATOM (CONCAT "SAVE." (CAR EACH))
                              " Format"))
  (SETQ SHOWTIME.MENU)])

```

**(SHOWTIME.SETUP.WINDOWPROPS**

```

[LAMBDA (WINDOW BITMAP NAME) (* TBigham "27-Mar-86 10:41")
  (PROGN (WINDOWPROP WINDOW 'TITLE (CONCAT "Viewer of " NAME))
         (WINDOWPROP WINDOW 'BITMAP.NAME NAME)
         (WINDOWPROP WINDOW 'BITMAP BITMAP])

```

**(SHOWTIME.SHOW.BITMAP**

```

[LAMBDA (BITMAP NAME WINDOW OPTION) (* TBigham " 7-Feb-87 18:43")
  (DECLARE (GLOBALVARS PROMPTWINDOW))
  (LET NIL (if (BITMAPP BITMAP)
    then [OR NAME (SETQ NAME (WINDOWPROP WINDOW 'BITMAP.NAME))
           (if (EQ OPTION 'RESHAPE&SHOW)
               then (SHOWTIME.RESHAPE.WINDOW BITMAP WINDOW)
               (CLEARW WINDOW)
               (BITBLT BITMAP 0 0 WINDOW)
               (SHOWTIME.SETUP.WINDOWPROPS WINDOW BITMAP NAME)
           else (printout PROMPTWINDOW T BITMAP " not a bitmap")])

```

**(SHOWTIME.WRITEBM**

```

[LAMBDA (BITMAP STREAM) (* TBigham " 1-Mar-87 18:19")
  (* "TBigham & Wall" " 7-Apr-86 10:19")
  (LET* ((BASE (fetch BITMAPBASE of BITMAP))
         (HEIGHT (fetch BITMAPHEIGHT of BITMAP))
         (WIDTH (fetch BITMAPWIDTH of BITMAP))
         (RWIDTH (fetch BITMAPPRASTERWIDTH of BITMAP))
         BOUNDARY BM)
    (if (ZEROP (REMAINDER WIDTH 32))
      then [for Y from 0 to (SUB1 HEIGHT) do (for X from 0 to (SUB1 RWIDTH)
        do (\BOUT STREAM (\GETBASEBYTE BASE 0))
           (\BOUT STREAM (\GETBASEBYTE BASE 1))
           (SETQ BASE (\ADDBASE BASE 1])
      else (SETQ BOUNDARY (for NUMBER from 32 by 32 thereis (if (ILEQ WIDTH NUMBER)
        then NUMBER)))
      (SETQ BM (BITMAPCREATE BOUNDARY HEIGHT))
      (BITBLT BITMAP 0 0 BM)
      (SHOWTIME.WRITEBM BM STREAM])

```

(RPAQQ SHOWTIME.ICON



```
(RPAQQ SHOWTIME.MASK

)
```

(RPAQQ SHOWTIME.LOAD.SUBITEMS NIL)

(RPAQQ SHOWTIME.SAVE.SUBITEMS NIL)

(RPAQQ SHOWTIME.MENU NIL)

(RPAQQ SHOWTIMETITLEREGION (7 7 56 29))

(RPAQQ SHOWTIME.DEFAULT.FORMAT LISP)

(RPAQQ BackgroundMenu NIL)

```
(RPAQQ SHOWTIME.FORMAT.FNS (SHOWTIME.FORMAT.FNS (RES READ.RES SHOWTIME.MAKE.RES)
(LISP SHOWTIME.READ.LISPBM SHOWTIME.SAVE.LISPBM)
(DIF SHOWTIME.LOAD.DIF.FILE NIL)
(BRUSH SHOWTIME.LOAD.BRUSH MAKEBRUSH)
(PRESS READPRESS PRESSBITMAP)))
```

```
(APPENDTOVAR BackgroundMenuCommands (Showtime ' (SHOWTIME)
"Opens a showtime window for use."))
```

(FILESLOAD BITMAPFNS SCALEBITMAP READBRUSH)

(SHOWTIME.ADD.FORMAT)

(PUTPROPS SHOWTIME COPYRIGHT ("Xerox Corporation" 1986 1987 1988 1989))



---

**FUNCTION INDEX**

GET.SHOWTIME.MENU .....	1	SHOWTIME.LOAD.BITMAP .....	4	SHOWTIME.READ.RES .....	6
INFORES .....	2	SHOWTIME.LOAD.BRUSH .....	4	SHOWTIME.RES.CHECK&MESSAGE .....	6
MAKEBRUSH .....	1	SHOWTIME.LOAD.DIF.FILE .....	4	SHOWTIME.RESHAPE.WINDOW .....	6
MAKEBRUSH.HEADER&BITMAP .....	2	SHOWTIME.LOAD.RES.FILE .....	4	SHOWTIME.SAVE.BITMAP .....	6
READ.RES .....	2	SHOWTIME.MAKE.RES .....	4	SHOWTIME.SAVE.LISPBM .....	6
SHOWTIME .....	2	SHOWTIME.MAKE.RES.HEADER .....	5	SHOWTIME.SCALE.BITMAP .....	6
SHOWTIME.ADD.FORMAT .....	7	SHOWTIME.MAKE.RES.TAIL .....	5	SHOWTIME.SETUP.WINDOWPROPS .....	7
SHOWTIME.BUTTONEVENTFN .....	3	SHOWTIME.READ.BRUSH .....	5	SHOWTIME.SHOW.BITMAP .....	7
SHOWTIME.GET.NAME .....	3	SHOWTIME.READ.LISPBM .....	6	SHOWTIME.WRITEBM .....	7
SHOWTIME.ICONFN .....	3	SHOWTIME.READ.PRESS .....	6		

---

**VARIABLE INDEX**

BackgroundMenu .....	8	SHOWTIME.ICON .....	7	SHOWTIME.MENU .....	8
SHOWTIME.DEFAULT.FORMAT .....	8	SHOWTIME.LOAD.SUBITEMS .....	8	SHOWTIME.SAVE.SUBITEMS .....	8
SHOWTIME.FORMAT.FNS .....	8	SHOWTIME.MASK .....	8	SHOWTIMETITLEREGION .....	8

---