File created: 16-May-90 18:16:27 **{DSK}<usr>local>lde>lispcore>sources>ICONW.;2**

changes to: (VARS ICONWCOMS)

previous date: 17-Dec-87 17:45:42 {DSK}<usr>local>lde>lispcore>sources>ICONW.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
```
;; Copyright (c) 1984, 1985, 1986, 1987, 1990 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ ICONWCOMS
       (;; Support for icons with non-rectangluar shape and with custom lettering inside them.
        (FNS ICONW ICONW.SHADE \ICONW.REPAINTFN \ICONW.COPYBUTTONEVENTFN)
        (FNS TITLEDICONW ICONW.TITLE \ICONW.SHOW.TITLE ICONW.PRINT-JUSTIFIED \ICONW.FORMAT.TITLE
             \ICONW.FORMAT.TITLE1 ICONTITLE)
        (COMS                                                    ; for use as DEFAULTICONFN
              (FNS TEXTICON)
              (INITVARS (DEFAULTTEXTICON)))
        (RECORDS TITLEDICON)
        (INITVARS (DEFAULTICONWIDTH 100)
               (DEFAULTICONFONT (FONTCREATE 'HELVETICA 10)))
        (DECLARE%: DONTCOPY (RECORDS ICONTITLE)
               (GLOBALVARS DEFAULTICONWIDTH DEFAULTICONFONT DEFAULTTEXTICON WBorder)
               (CONSTANTS (ICONSELECTIONSHADE 23130))
               (MACROS .COPYKEYDOWNP.))))
```

;; Support for icons with non-rectangluar shape and with custom lettering inside them.

```
(DEFINEQ
```

(ICONW

```
  [LAMBDA (ICON MASK POSITION NOOPENFLG)                         (* bvm%: "26-Aug-85 16:01")
```

;; creates a window that merges with its background. This is done by putting the background in the original bits, erasing the bits that are on in ;; MASK and then painting the bits from IMAGEBM.

```
    [COND
       ((NOT (type? POSITION POSITION))
        (SETQ POSITION (GETBOXPOSITION (fetch (BITMAP BITMAPWIDTH) of ICON)
                                       (fetch (BITMAP BITMAPHEIGHT) of ICON]
    (LET ((ICONW (CREATEW (create REGION
                                  LEFT _ (fetch (POSITION XCOORD) of POSITION)
                                  BOTTOM _ (fetch (POSITION YCOORD) of POSITION)
                                  WIDTH _ (fetch (BITMAP BITMAPWIDTH) of ICON)
                                  HEIGHT _ (fetch (BITMAP BITMAPHEIGHT) of ICON))
                          NIL 0 T)))
         (WINDOWPROP ICONW 'RESHAPEFN 'DON'T)
         (WINDOWPROP ICONW 'ICONIMAGE ICON)
         (WINDOWPROP ICONW 'ICONMASK MASK)
         (WINDOWPROP ICONW 'AFTERMOVEFN (FUNCTION \ICONW.REPAINTFN))
         (WINDOWPROP ICONW 'TOTOPFN (FUNCTION \ICONW.REPAINTFN))
         (WINDOWPROP ICONW 'OPENFN (FUNCTION \ICONW.REPAINTFN))
         (OR NOOPENFLG (OPENW ICONW))
         ICONW])
```

(ICONW.SHADE

```
  [LAMBDA (WINDOW SHADE)                                         ; Edited 20-Feb-87 11:02 by jds
    (AND (WINDOWP WINDOW)
         (LET (SHADEBM ERASEBM IMAGEBM)
              [COND
                 (SHADE (COND
                           [(NEQ SHADE WHITESHADE)               ; Build an auxiliary bitmap that is shaded the requested shade in
                                                                 ; all the parts where the image shows
                            [OR (SETQ SHADEBM (WINDOWPROP WINDOW 'SHADEIMAGE))
                                (WINDOWPROP WINDOW 'SHADEIMAGE (SETQ SHADEBM (BITMAPCREATE
                                                                  [fetch (BITMAP BITMAPWIDTH)
                                                                     of (SETQ IMAGEBM
                                                                          (WINDOWPROP WINDOW
                                                                                 'ICONIMAGE]
                                                                  (fetch (BITMAP BITMAPHEIGHT)
                                                                     of IMAGEBM]
                            (BLTSHADE SHADE SHADEBM 0 0 NIL NIL 'REPLACE)
                            (COND
                               ((SETQ ERASEBM (WINDOWPROP WINDOW 'ICONMASK))
                                (BITBLT ERASEBM 0 0 SHADEBM 0 0 NIL NIL 'INVERT 'ERASE]
                           (T (WINDOWPROP WINDOW 'SHADEIMAGE NIL]
              (PROG1 (WINDOWPROP WINDOW 'ICONSHADE SHADE)
                  (AND SHADE (OPENWP WINDOW)
                       (\ICONW.REPAINTFN WINDOW)))])
```

(\ICONW.REPAINTFN

```
  [LAMBDA (WINDOW)                                          (* bvm%: "31-Jul-85 14:05")
    (PROG ((IMAGEBM (WINDOWPROP WINDOW 'ICONIMAGE))
           (ERASEBM (WINDOWPROP WINDOW 'ICONMASK))
           (SHADEBM (WINDOWPROP WINDOW 'SHADEIMAGE))
           WIDTH HEIGHT)
          (SETQ WIDTH (fetch (BITMAP BITMAPWIDTH) of IMAGEBM))
          (SETQ HEIGHT (fetch (BITMAP BITMAPHEIGHT) of IMAGEBM))
          (TOTOPW WINDOW T)                                 ; Bring the window to the top without calling its TOTOPFN (i.e.,
                                                            ; this very fn)
          [COND
             (ERASEBM
                    ;; There's clipping to do, so copy the background, erase bits where the image lies, then OR in the image
                    (BITBLT (WINDOWPROP WINDOW 'IMAGECOVERED)
                           0 0 WINDOW 0 0 WIDTH HEIGHT 'INPUT 'REPLACE)
                    (BITBLT ERASEBM 0 0 WINDOW 0 0 WIDTH HEIGHT 'INPUT 'ERASE)
                    (BITBLT IMAGEBM 0 0 WINDOW 0 0 WIDTH HEIGHT 'INPUT 'PAINT))
             (T                                             ; No clipping, just copy out the original image
                (BITBLT IMAGEBM 0 0 WINDOW 0 0 WIDTH HEIGHT 'INPUT 'REPLACE]
          (COND
             (SHADEBM                                       ; The image is to be shaded
                    (BITBLT SHADEBM 0 0 WINDOW 0 0 WIDTH HEIGHT 'INPUT 'PAINT])
```

(\ICONW.COPYBUTTONEVENTFN

```
  [LAMBDA (WINDOW)                                          ; Edited 17-Dec-87 17:42 by bvm:
```

;;; copy select the window's title (or whatever it says to copy).

```
    (PROG* ((TITLESPEC (WINDOWPROP WINDOW 'ICONTITLESPEC))
            (REG (fetch ICREGION of TITLESPEC))
            (LEFT (fetch (REGION LEFT) of REG))
            (BOTTOM (fetch (REGION BOTTOM) of REG))
            (WIDTH (fetch (REGION WIDTH) of REG))
            (HEIGHT (fetch (REGION HEIGHT) of REG))
            (SHADE ICONSELECTIONSHADE)
            (SELECTEDP T)
            COPYFN)
       SELECTEDLP
            (TOTOPW WINDOW)                                 ; Draw a box around the title region
            (BLTSHADE SHADE WINDOW LEFT BOTTOM WIDTH 2 'INVERT)
            (BLTSHADE SHADE WINDOW LEFT (+ BOTTOM 2)
                   2
                   (- HEIGHT 4)
                   'INVERT)
            (BLTSHADE SHADE WINDOW LEFT (+ BOTTOM HEIGHT -2)
                   WIDTH 2 'INVERT)
            (BLTSHADE SHADE WINDOW (+ LEFT WIDTH -2)
                   (+ BOTTOM 2)
                   2
                   (- HEIGHT 4)
                   'INVERT)
       LP   (if (NEQ SELECTEDP (SETQ SELECTEDP (EQ (WHICHW)
                                                   WINDOW)))
                then (if SELECTEDP
                         then                               ; Moved back inside
                              (GO SELECTEDLP)
                       else                                 ; Moved outside
                            (\ICONW.REPAINTFN WINDOW)))     ; wait for a button up or move out of region
       LP2 (BLOCK)
            (COND
               ((NOT (.COPYKEYDOWNP.))                      ; Finished, copy selected item
                [COND
                   (SELECTEDP (\ICONW.REPAINTFN WINDOW)
                          (if (SETQ COPYFN (WINDOWPROP WINDOW 'COPYFN))
                              then                          ;  Window says how to copy select
                                   (CL:FUNCALL COPYFN WINDOW)
                            else (BKSYSBUF (fetch ICTITLE of TITLESPEC]
                (until (MOUSESTATE UP) do (BLOCK))          ; Wait for mouse to come up, so we don't get bogus button event
                                                            ; afterwards
                (RETURN))
               ((MOUSESTATE UP)
                (if (NOT SELECTEDP)
                    then                                    ; Button up, and outside window, so can return
                         (RETURN)
                  else                                      ; Wait for copy to come up
                       (GO LP2)))
               (T                                           ; While button is down, watch where mouse is
                  (GO LP])

)

(DEFINEQ
```

(TITLEDICONW

```
[LAMBDA (ICON TITLE FONT POSITION NOOPENFLG JUST BREAKCHARS OPERATION)
                                                        ; Edited 17-Dec-87 17:22 by bvm:
    ;; Given a TITLEDICON, create an instance of it with specific text.
    (LET (BITS ICONW TITLESPEC REG MASK FORMATTED)
         [COND
            [(NOT BREAKCHARS)
             (SETQ BREAKCHARS (CHARCODE (SPACE]
            [(EQ BREAKCHARS 'FILE)                      ; File name field separators
             (SETQ BREAKCHARS (CHARCODE (SPACE - } %: > %. ; /]
            ((NLISTP BREAKCHARS)
             (SETQ BREAKCHARS (LIST BREAKCHARS]
         (SETQ FONT (FONTCREATE (OR FONT DEFAULTICONFONT)))
         (SELECTQ OPERATION
              ((REPLACE INVERT))
              (ERASE (SETQQ OPERATION INVERT))
              ((NIL PAINT)
                   (SETQQ OPERATION REPLACE))
              (\ILLEGAL.ARG OPERATION))
         [COND
            (ICON (SETQ BITS (BITMAPCOPY (fetch (TITLEDICON ICON) of ICON)))
                  (SETQ REG (fetch TITLEREG of ICON))
                  (SETQ MASK (fetch (TITLEDICON MASK) of ICON)))
            (T (LET ((TITLEWIDTH (STRINGWIDTH TITLE FONT))
                     (BORDER WBorder)
                     WIDTH HEIGHT)                      ; Make a simple rectangle with a border like a window
                    (SETQ FORMATTED (\ICONW.FORMAT.TITLE TITLE FONT (IMAX DEFAULTICONWIDTH (LRSH TITLEWIDTH 1))
                                                BREAKCHARS))
                    ;; Try actually formatting the title, expecting about three lines, to see what dimensions the window needs to be
                    (SETQ WIDTH (WIDTHIFWINDOW (OR (CDR (for X in FORMATTED largest (CDR X)))
                                                   DEFAULTICONWIDTH)
                                               BORDER))
                    (SETQ HEIGHT (HEIGHTIFWINDOW (TIMES (LENGTH FORMATTED)
                                                        (FONTPROP FONT 'HEIGHT))
                                                 NIL BORDER))
                    (SETQ BITS (BITMAPCREATE WIDTH HEIGHT))
                    (BLTSHADE BLACKSHADE BITS 0 0 WIDTH HEIGHT 'REPLACE)
                                                        ; Fill with black, then white out everything but the border
                    [COND
                       ((NEQ OPERATION 'INVERT)
                        (BLTSHADE WHITESHADE BITS (FOLDHI BORDER 2)
                                (FOLDHI BORDER 2)
                                (- WIDTH BORDER)
                                (- HEIGHT BORDER)
                                'REPLACE]
                    (SETQ REG (CREATEREGION BORDER BORDER (- WIDTH (LLSH BORDER 1))
                                            (- HEIGHT (LLSH BORDER 1]
         (SETQ ICONW (ICONW BITS MASK POSITION T))
         (WINDOWPROP ICONW 'ICONTITLESPEC
                (SETQ TITLESPEC
                 (create ICONTITLE
                        ICIMAGE _ BITS
                        ICFONT _ FONT
                        ICJUST _ JUST
                        ICREGION _ REG
                        ICBREAKCHARS _ BREAKCHARS
                        ICOPERATION _ OPERATION)))
         (WINDOWPROP ICONW 'COPYBUTTONEVENTFN (FUNCTION \ICONW.COPYBUTTONEVENTFN))
         (\ICONW.SHOW.TITLE ICONW TITLESPEC (MKSTRING (OR TITLE " "))
                T)                                      ; Create a copy of the icon image, with the text imposed on it.
                                                        ; Save it for restoration on open, repaint, &c
         (OR NOOPENFLG (OPENW ICONW))                   ; Open the window, unless he wants it kept closed.
         ICONW])
```

(ICONW.TITLE

```
[LAMBDA (ICONW TITLE)                                   (* bvm%: "30-Aug-85 17:19")
```

;;; Returns current title of icon, sets new title if TITLE not NIL

```
    (LET [(TITLESPEC (WINDOWPROP ICONW 'ICONTITLESPEC]
         (COND
            ((NOT TITLESPEC)
             (ERROR "Not a titled icon" ICONW))
            (T (PROG1 (fetch ICTITLE of TITLESPEC)
                   [COND
                      (TITLE (\ICONW.SHOW.TITLE ICONW TITLESPEC TITLE)
                             (AND (OPENWP ICONW)
                                  (\ICONW.REPAINTFN ICONW]) ])
```

(\ICONW.SHOW.TITLE

```
[LAMBDA (ICONW TITLESPEC TEXT NEWFLG)                   ; Edited 17-Dec-87 15:35 by bvm:
```

;; Create a copy of the icon window's image bitmap, complete with text in place according to the TITLESPEC. If NEWFLG, don't bother erasing the ;; text area.

```
    (LET [(JUST (fetch ICJUST of TITLESPEC))
          (FONT (fetch ICFONT of TITLESPEC))
          (REG (fetch ICREGION of TITLESPEC))
          (BITS (BITMAPCOPY (fetch ICIMAGE of TITLESPEC)))
          (OPERATION (fetch ICOPERATION of TITLESPEC))
          (MASK (WINDOWPROP ICONW 'ICONMASK]                    ; Set up a displaystream so we can print onto the icon's image
                                                                ; bitmap
         (ICONW.PRINT-JUSTIFIED (DSPCREATE BITS)
                JUST FONT REG OPERATION (fetch ICBREAKCHARS of TITLESPEC)
                TEXT NEWFLG)
         [COND
            (MASK (BITBLT MASK 0 0 BITS 0 0 (fetch BITMAPWIDTH of BITS)
                          (fetch BITMAPHEIGHT of BITS)
                          'INVERT
                          'ERASE]
         (replace ICTITLE of TITLESPEC with TEXT)
         (WINDOWPROP ICONW 'ICONIMAGE BITS)
         ICONW])
```

(ICONW.PRINT-JUSTIFIED

```
  [LAMBDA (STREAM JUST FONT REG OPERATION BREAKCHARS TEXT NEWFLG)
                                                                ; Edited 10-Nov-87 12:41 by jds
    (LET ((OLDCLIP (DSPCLIPPINGREGION REG STREAM))
          (REAL-FONT (FONTCOPY FONT 'DEVICE STREAM))
          LMARG MAXHEIGHT MAXWIDTH MAXLINES WIDTH FORMATTEDLINES FONTHEIGHT TITLEHEIGHT)
         (DSPFONT REAL-FONT STREAM)                             ; Set the right font
         (DSPOPERATION OPERATION STREAM)                        ; Don't erase any bits from the icon image--paint the msg
         (LINELENGTH 32000 STREAM)                              ; Avoid trouble with PRIN1
         (DSPLEFTMARGIN (SETQ LMARG (fetch (REGION LEFT) of REG))
                STREAM)                                         ; Left margin for the message
         (DSPRIGHTMARGIN 32700 STREAM)
         (COND
            ((NOT NEWFLG)                                       ; Clear anything in the title region
             (DSPFILL REG (SELECTQ OPERATION
                                (INVERT BLACKSHADE)
                                WHITESHADE)
                    'REPLACE STREAM)))
         (SETQ FONTHEIGHT (FONTPROP REAL-FONT 'HEIGHT))         ; Single line's height
         (SETQ MAXHEIGHT (fetch (REGION HEIGHT) of REG))        ; Max height of the title
         (SETQ MAXWIDTH (fetch (REGION WIDTH) of REG))
         [SETQ FORMATTEDLINES (\ICONW.FORMAT.TITLE TEXT REAL-FONT MAXWIDTH BREAKCHARS (SETQ MAXLINES
                                                                                            (IQUOTIENT MAXHEIGHT
                                                                                                   FONTHEIGHT]
         (SETQ TITLEHEIGHT (ITIMES FONTHEIGHT (FLENGTH FORMATTEDLINES)))
                                                                ; Height of the message
         (MOVETO LMARG [IPLUS (fetch (REGION BOTTOM) of REG)
                              (IDIFFERENCE [IMIN MAXHEIGHT (COND
                                                             ((EQMEMB 'TOP JUST)
                                                              ; Top-flush title
                                                              (fetch (REGION TOP) of REG))
                                                             ((EQMEMB 'BOTTOM JUST)
                                                              ; Bottom-flush title
                                                              (IPLUS (fetch (REGION BOTTOM) of REG)
                                                                     TITLEHEIGHT))
                                                             ((IGREATERP TITLEHEIGHT MAXHEIGHT)
                                                              MAXHEIGHT)
                                                             (T
                                                              ; Centered vertically title
                                                              (IDIFFERENCE MAXHEIGHT (LRSH (IDIFFERENCE
                                                                                             MAXHEIGHT
                                                                                             TITLEHEIGHT
                                                                                             )
                                                                                           1]
                                           (FONTPROP REAL-FONT 'ASCENT]
                STREAM)                                         ; Move to the left end of the first message line
         (bind (NCH _ 0) to MAXLINES as LINE in FORMATTEDLINES
            do                                                  ; FORMATTEDLINES is a list of elements (lastch# . width)
               [COND
                  ((NOT (EQMEMB 'LEFT JUST))                    ; Move to this line's left end
                   (LET [(LEFTOVER (IDIFFERENCE MAXWIDTH (CDR LINE]
                        (RELMOVETO (COND
                                     ((EQMEMB 'RIGHT JUST)
                                      LEFTOVER)
                                     (T (LRSH LEFTOVER 1)))
                                   0 STREAM]
               (bind (MAXCHAR _ (CAR LINE))
                     CH do                                      ; Print the characters -- except the final SPACE on a line, or a
                                                                ; CR
                     (SETQ CH (NTHCHARCODE TEXT (add NCH 1)))
                     (COND
                        ([NOT (AND (EQ NCH (CAR LINE))
                                   (FMEMB CH (CHARCODE (CR SPACE]
                         (\OUTCHAR STREAM CH)))
                  repeatuntil (EQ NCH MAXCHAR))
               (TERPRI STREAM))
         (DSPCLIPPINGREGION OLDCLIP STREAM])
```

(\ICONW.FORMAT.TITLE

```
[LAMBDA (TITLE FONT MAXWIDTH BREAKCHARS MAXLINES)            (* bvm%: "27-Aug-85 18:21")
  (LET ((RESULT (\ICONW.FORMAT.TITLE1 TITLE FONT MAXWIDTH BREAKCHARS)))
       (COND
          ((OR (NULL MAXLINES)
               (GEQ MAXLINES (LENGTH RESULT)))                ; It fit, so return it
           RESULT)
          (T                                                  ; Try breaking less
             (LET ((WASTED 0)
                   (EXCESS 0))
                  [for I from 1 as LINE in RESULT do (COND
                                                       [(LEQ I MAXLINES)
                                                        (add WASTED (IDIFFERENCE MAXWIDTH (CDR LINE]
                                                       (T (add EXCESS (CDR LINE]
                  (COND
                     ([AND (LESSP EXCESS WASTED)
                           (GEQ MAXLINES (LENGTH (SETQ RESULT (\ICONW.FORMAT.TITLE1 TITLE FONT MAXWIDTH
                                                                BREAKCHARS (IDIFFERENCE
                                                                            MAXWIDTH
                                                                            (IQUOTIENT (IDIFFERENCE
                                                                                        WASTED
                                                                                        EXCESS)
                                                                                       MAXLINES]
                                                                                       ; Reformatted okay by forcing less wastage per line
                      RESULT)
                     (T                                       ; Take out all the breaks
                        (\ICONW.FORMAT.TITLE1 TITLE FONT MAXWIDTH NIL MAXWIDTH])
```

(\ICONW.FORMAT.TITLE1

```
[LAMBDA (TITLE FONT MAXWIDTH BREAKCHARS MINWIDTH)            ; Edited  9-Nov-87 14:01 by bvm:
  (LET* ((TITLELEN (NCHARS TITLE))
         (DONE (EQ TITLELEN 0))
         (FONTHEIGHT (FONTPROP FONT 'HEIGHT))
         (NCH 0)
         (WIDTHSOFAR 0)
         TCH CHWIDTH)
        (until DONE
           collect                                            ; Gather the icon title, broken into lines which fit.
                  (bind CH LASTBREAKPOS LASTBREAKWIDTH
                     do                                      ; Run thru the characters one by one.
                         (COND
                            ((>= NCH TITLELEN)               ; That was the last character.
                             (SETQ DONE T)
                             (RETURN (CONS TITLELEN WIDTHSOFAR)))
                            (T                               ; Look at the next character.
                               (SETQ CH (NTHCHARCODE TITLE (add NCH 1)))
                               [COND
                                  ((EQ CH (CHARCODE CR))     ; CR forces a new line.
                                   (RETURN (PROG1 (CONS NCH WIDTHSOFAR)
                                                  (SETQ WIDTHSOFAR 0]
                               [COND
                                  ((IGREATERP (add WIDTHSOFAR (SETQ CHWIDTH (CHARWIDTH CH FONT)))
                                          MAXWIDTH)          ; We're past the right margin.  Time to stop.
                                   (RETURN (COND
                                              [(AND (EQ CH (CHARCODE SPACE))
                                                    (FMEMB CH BREAKCHARS))
                                                                ; We just happened to end at a space, so it's safe to break here
                                               (PROG1 (CONS NCH (- WIDTHSOFAR CHWIDTH))
                                                   (COND
                                                      ((EQ NCH TITLELEN)
                                                                ; Title ends in a space, which we will not print, so we're done
                                                                ; (otherwise, we'd have nothing to collect for the next line).
                                                       (SETQ DONE T))
                                                      (T        ; Since we're breaking exactly here, we have nothing leftover for
                                                                ; the next line.
                                                         (SETQ WIDTHSOFAR 0))))]
                                              [LASTBREAKPOS     ; There is a space we can break the line at.  Break there.
                                                     (SETQ WIDTHSOFAR (- WIDTHSOFAR LASTBREAKWIDTH))
                                                     (CONS LASTBREAKPOS (COND
                                                                           ((EQ (NTHCHARCODE TITLE LASTBREAKPOS)
                                                                                (CHARCODE SPACE))
                                                                            (- LASTBREAKWIDTH
                                                                               (CHARWIDTH (CHARCODE SPACE)
                                                                                          FONT)))
                                                                           (T LASTBREAKWIDTH]
                                              (T                ; There were no spaces on this line.  Break after the last
                                                                ; character that did fit.
                                                 (CONS (SUB1 NCH)
                                                       (- WIDTHSOFAR (SETQ WIDTHSOFAR CHWIDTH]
                               (COND
                                  ((AND (FMEMB CH BREAKCHARS)
                                        (OR (NULL MINWIDTH)
                                            (>= WIDTHSOFAR MINWIDTH)))
                                                                ;; Remember where spaces are, so we can back up and split lines there if possible.  Don't split if there
```

```
                                    ;; isn't enough on the line yet
                                    (SETQ LASTBREAKPOS NCH)
                                    (SETQ LASTBREAKWIDTH WIDTHSOFAR])
```

(ICONTITLE

```
  [LAMBDA (MSG REG FONT ICONW JUST)                                    (* bvm%: "16-Aug-85 20:20")
                                                                       ; Obsolete entry
    (LET [(TITLESPEC (WINDOWPROP ICONW 'ICONTITLESPEC]
         (COND
            ((NOT TITLESPEC)
             (ERROR "Not a titled icon" ICONW))
            (T (COND
                  (REG (replace ICREGION of TITLESPEC with REG)))
               (COND
                  (FONT (replace ICFONT of TITLESPEC with FONT)))
               (COND
                  (JUST (replace ICJUST of TITLESPEC with JUST)))
               (\ICONW.SHOW.TITLE ICONW TITLESPEC MSG])

)

;; for use as DEFAULTICONFN

(DEFINEQ

## (TEXTICON

  [LAMBDA (WINDOW TEXT)                                                (* bvm%: "18-Mar-86 16:54")
    (OR (WINDOWP TEXT)
        (LET* [[ICON (TITLEDICONW DEFAULTTEXTICON [COND
                                                  (TEXT)
                                                  ((AND (SETQ TEXT (WINDOWPROP WINDOW 'TITLE))
                                                        (NEQ (NCHARS TEXT)
                                                             0))
                                                   TEXT)
                                                  (T (CONCAT "Icon made " (DATE]
                           NIL
                           (WINDOWPROP WINDOW 'ICONPOSITION]
              (REG (WINDOWPROP ICON 'REGION]
             (WINDOWPROP WINDOW 'ICONPOSITION (create POSITION
                                                  XCOORD _ (fetch (REGION LEFT) of REG)
                                                  YCOORD _ (fetch (REGION BOTTOM) of REG)))
                                                   ; Remember position for the next shrinkage
              ICON])

)

(RPAQ? DEFAULTTEXTICON )

(DECLARE%: EVAL@COMPILE

(RECORD TITLEDICON (ICON MASK TITLEREG))
)

(RPAQ? DEFAULTICONWIDTH 100)

(RPAQ? DEFAULTICONFONT (FONTCREATE 'HELVETICA 10))

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD ICONTITLE (ICIMAGE ICTITLE ICFONT ICJUST ICREGION ICBREAKCHARS ICOPERATION))
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DEFAULTICONWIDTH DEFAULTICONFONT DEFAULTTEXTICON WBorder)
)

(DECLARE%: EVAL@COMPILE

(RPAQQ ICONSELECTIONSHADE 23130)

(CONSTANTS (ICONSELECTIONSHADE 23130))
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS .COPYKEYDOWNP. MACRO [NIL (OR (KEYDOWNP 'LSHIFT)
                                        (KEYDOWNP 'RSHIFT)
                                        (KEYDOWNP 'COPY])
)
)

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

FUNCTION INDEX

VARIABLE INDEX

RECORD INDEX

MACRO INDEX

CONSTANT INDEX