

File created: 28-May-86 17:17:07 {ICE}<DENBER>LISP>ACE-EDIT.;6

changes to: (FNS ACE.COMPILE.FRAME.ACE ACE.EDIT.PAINT ACE.EXTRACTBM ACE.EDIT.SETUP.EDIT.MENU ACE.EDIT.FRAME  
ACEROTATEREGION ROTATEBM ACEEDITBM)  
(VARS ACE-EDITCOMS ACELOGOMAP)

previous date: 16-May-86 14:41:14 {ICE}<DENBER>LISP>ACE-EDIT.;4

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(\* \* Copyright (c) 1986 by Xerox Corporation. All rights reserved.)

(RPAQQ ACE-EDITCOMS

```
(( * TOP LEVEL EDITING STUFF)
(FNS ACE.EDIT.ACE.EDIT.FRAME ACE.EDIT.SETUP.EDIT.MENU ACE.GETREGIONFACTOR ACEROTATEREGION ACESCALEREGION)
(* LINEART FNS)
(FNS ACE.EDIT.LINEART ACE.EDIT.LINEART.DRAW ACE.EDIT.LINEART.ADJ ACE.EDIT.LINEART.TRACKLINE)
(* OTHER EDITING STUFF)
(FNS ACE.EDIT.MOVE.REGION ACE.EDIT.COMBINE.REGION ACE.EDIT.TEXT ACE.EDIT.TEXTURE.REGION
ACE.EDIT.TEXTURE.AREA ACE.EDIT.PAINT ACE.FILLWITHTEXTURE ACE.SCANLINESEEDFILL
ACE.EDIT.CREATE.MENU.TEXTURES ACE.EDIT.PUTDOWN.BITMAP ACE.EDIT.MOVE.REGION.ASST ACEEDITBM
ACE.READBRUSHSHAPE)
(* TABLET AND SUPPORT FNS)
(FNS ACE.EDIT.POINT&CODE ACE.GET.DEVICE.STATE ACE.GET.DEVICE.STATE&CURSOR ACE.EXTRACTBM
ACE.EDIT.REDRAW.ABITMAP ACE.SCALE.BITMAP ACE.COMPILE.FRAME.ACE ACE.MM1201.INIT ACE.MM1201POLL
ACE.MM1201.PROBLEM ACE.EDIT.CLEAR.ALL.MENUS ROTATEBM SIGN TEXTURELINE \TEXTURELINE RS232LOSTCHARFN)
(MACROS ACE.POPPOS ACE.PUSHPOS)
(CURSORS ACE.EDIT.LINEART.SQUARE.CURSOR)
(BITMAPS ACELOGOMAP)
(VARS (RS232LOSTCHARFN (QUOTE RS232LOSTCHARFN)))
(* MENUS FOR ACE-EDIT)
(GLOBALVARS ACE.CONTROL.WINDOW ACE.EDIT.FRAME.MENU ACE.EDIT.LINEART.ADJ.MENU ACE.EDIT.MOVE.MENU
ACE.EDIT.TEXT.FONT.MENU ACE.EDIT.TEXT.SIZE.MENU ACE.EDIT.TEXT.FACE.MENU ACE.EDIT.TEXTURE.MENU
ACE.EDIT.PUTDOWN.MENU)
(P (ACE.EDIT.CLEAR.ALL.MENUS)))
```

(\* \* TOP LEVEL EDITING STUFF)

(DEFINEQ

(ACE.EDIT

[LAMBDA (FLG)

(\* PmT "24-Apr-85 14:51")

(\* Entry into frame editing. Reconstructs CURRENT frame, previous  
(if any); and successor (if any); Calls ACE.EDIT.FRAME on the reconstructed CURRENT frame  
(a bitmap); when editing is complete, recompiles current frame with previous and successor)

```
(PROG (BEFORE.BM PRESENT.BM AFTER.BM)
(ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
"EDIT")
(COND
([NULL (AND (ACE.CHECKSTUFF (QUOTE SEQ))
(ACE.CHECKSTUFF (QUOTE FRAME]
NIL)
(T [COND
((EQ ACE.CURRENT.FRAME ACE.CURRENT.SEQUENCE)
(SETQ BEFORE.BM NIL))
(T (SETQ BEFORE.BM (ACE.RECONSTRUCT.FRAME (LDIFF ACE.CURRENT.SEQUENCE ACE.CURRENT.FRAME]
[COND
[BEFORE.BM (SETQ PRESENT.BM (ACE.RECONSTRUCT.FRAME (LDIFF ACE.CURRENT.SEQUENCE ACE.FRAME.TAIL
1
(T (SETQ PRESENT.BM (fetch (ACE.BLIT BITMAP) of (CAR (fetch (ACE.FRAME BLITS)
of (CAR ACE.CURRENT.SEQUENCE]
[COND
((NULL ACE.FRAME.TAIL)
(SETQ AFTER.BM NIL))
(T (SETQ AFTER.BM (ACE.RECONSTRUCT.FRAME (LDIFF ACE.CURRENT.SEQUENCE (CDR ACE.FRAME.TAIL]
[COND
((ACE.MAC.SEQ.CLIP (ACE.EDIT.FRAME PRESENT.BM) )
(COND
((NULL (OR BEFORE.BM AFTER.BM) )
NIL)
((NULL BEFORE.BM)
(replace (ACE.FRAME BLITS) of (CAR ACE.FRAME.TAIL) with (ACE.COMPILE.FRAME.ACE PRESENT.BM
AFTER.BM ACE.VERTICAL.BLOCK
ACE.AREA.THRESHOLD)))
((NULL AFTER.BM)
(replace (ACE.FRAME BLITS) of (CAR ACE.CURRENT.FRAME) with (ACE.COMPILE.FRAME.ACE
BEFORE.BM PRESENT.BM
ACE.VERTICAL.BLOCK
ACE.AREA.THRESHOLD)))
(T (replace (ACE.FRAME BLITS) of (CAR ACE.CURRENT.FRAME) with (ACE.COMPILE.FRAME.ACE
```

```

BEFORE.BM PRESENT.BM
ACE.VERTICAL.BLOCK
ACE.AREA.THRESHOLD))
(replace (ACE.FRAME BLITS) of (CAR ACE.FRAME.TAIL) with (ACE.COMPILE.FRAME.ACE
PRESENT.BM AFTER.BM
ACE.VERTICAL.BLOCK
ACE.AREA.THRESHOLD]
(ACE.QUICKDRAW&UPD (LDIFF ACE.CURRENT.SEQUENCE ACE.FRAME.TAIL])

```

**(ACE.EDIT.FRAME**

[LAMBDA (ABITMAP)

(\* MJD "23-May-86 15:51")

(\* Does editing on ABITMAP (this suppose to be a frame; but not necessarily);  
 Loops through an options menu until QUIT is selected. The BITBLT just redraws the frame as it really is after every editing  
 option)

```

(PROG NIL
  LOOP
    (BITBLT ABITMAP 0 0 ACE.SEQ.WINDOW ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE)
      NIL NIL)
    (ACE.UPD.CONTROL.WINDOW (QUOTE FRAME)
      T)
    (SELECTQ (MENU (OR ACE.EDIT.FRAME.MENU (ACE.EDIT.SETUP.EDIT.MENU)))
      (NIL NIL)
      (PAINT (ACE.EDIT.PAINT ABITMAP))
      (EDITBM (ACE.EDIT.BM ABITMAP))
      (LINEART (ACE.EDIT.LINEART ABITMAP))
      (MOVE.REG (ACE.EDIT.MOVE.REGION ABITMAP))
      (COMBINE.REG (ACE.EDIT.COMBINE.REGION ABITMAP))
      (SNAPREG (SNAPW))
      (TEXT (ACE.EDIT.TEXT ABITMAP))
      (TEXTURE.AREA (ACE.EDIT.TEXTURE.AREA ABITMAP))
      (TEXTURE.REG (ACE.EDIT.TEXTURE.REGION ABITMAP))
      (CLEAR (ACE.EDIT.TEXTURE.REGION ABITMAP WHITESHADE))
      (SCALEREG (ACESCALEREGION (QUOTE SWEEP)))
      (SCALEREGXY (ACESCALEREGION (QUOTE XY)))
      (SCALEREGX (ACESCALEREGION (QUOTE X)))
      (SCALEREGY (ACESCALEREGION (QUOTE Y)))
      (ROTATEREG (ACEROTATEREGION))
      (2D (EDIT.SEQ.2D ABITMAP))
      (3D (EDIT.SEQ.3D ABITMAP))
      (QUIT (RETURN T))
      (ABORT (AND (ACE.CONFIRMIT "Click LEFT to Ignore Changes." T)
        (RETURN NIL)))
      NIL)
    (GO LOOP])

```

**(ACE.EDIT.SETUP.EDIT.MENU**

[LAMBDA NIL

(\* MJD "23-May-86 15:42")

```

(SETQ ACE.EDIT.FRAME.MENU (create MENU
  ITEMS _ (QUOTE ((Paint" (QUOTE PAINT)
    "Does standard Paint on frame")
    ("Line art" (QUOTE LINEART)
    "Enter lines through various devices")
    ("Text" (QUOTE TEXT)
    "Put text into frame")
    ("Edit bits" (QUOTE EDITBM)
    "Calls the Bitmap Editor on frame")
    (" " NIL "Just a Spacer")
    ("Move region" (QUOTE MOVE.REG)
    "Move a rectangular region in frame")
    ("Combine region" (QUOTE COMBINE.REG)
    "Combine any region on screen into frame")
    ("Snap region" (QUOTE SNAPREG)
    "Save a region in a scratch window")
    ["Scale region" (QUOTE SCALEREG)
    "Change size of a region in the frame"
    (SUBITEMS ("To a new region" (QUOTE SCALEREG)
      ("In x and y" (QUOTE SCALEREGXY))
      ("In x only" (QUOTE SCALEREGX))
      ("In y only" (QUOTE SCALEREGY))
    ("Rotate region" (QUOTE ROTATEREG)
    "Rotates a region any number of degrees")
    ("Texture region fill" (QUOTE TEXTURE.AREA)
    "Texture any closed area within frame")
    ("Texture box fill" (QUOTE TEXTURE.REG)
    "Fills a bounded rectangle with a texture.")
    ("Clear region" (QUOTE CLEAR)
    "Erases a specified Region")
    (" " NIL "Just a Spacer")
    ("Compile frame" (QUOTE QUIT)
    "Exits the editor and compiles the frame.")
    ("Quit - ABORT" (QUOTE ABORT)
    "Stops editor; frame reverts to original state.")))

```

```

CENTERFLG _ T
TITLE _ "Edit Options"
CHANGEOFFSETFLG _ T])

```

**(ACEGETREGIONFACTOR**

(\* MD "19-Jun-85 14:29")

```

[LAMBDA (W H)
  (PROG (TEMPREGION)
    (ACE.TELLEM "Select desired size for new region" T)
    (SETQ TEMPREGION (GETREGION NIL NIL NIL (QUOTE ACE.EDIT.MOVE.REGION.ASST)
      NIL))
    (RETURN (CONS (FQUOTIENT (fetch (REGION WIDTH) of TEMPREGION)
      W)
      (FQUOTIENT (fetch (REGION HEIGHT) of TEMPREGION)
      H)))

```

**(ACEROTATEREGION**

(\* MJD "23-May-86 15:15")

```

[LAMBDA (TYPE)
  (PROG (TEMPREGION TEMPBM NEWBM)
    (ACE.TELLEM "Select a Region inside the Sequence" T)
    (SETQ TEMPREGION (GETREGION NIL NIL NIL (QUOTE ACE.EDIT.MOVE.REGION.ASST)
      NIL))
    (SETQ TEMPBM (BITMAPCREATE (fetch (REGION WIDTH) of TEMPREGION)
      (fetch (REGION HEIGHT) of TEMPREGION)
      1))
    (BITBLT ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      TEMPBM 0 0 NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE))
    (SETQ NEWBM (BITMAPCREATE (fetch (REGION WIDTH) of TEMPREGION)
      (fetch (REGION HEIGHT) of TEMPREGION)
      1))
    (ROTATEBM TEMPBM NEWBM (RNUMBER "Angle in degrees:"))
    (BITBLT TEMPBM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      NIL NIL (QUOTE INPUT)
      (QUOTE ERASE))
    (BITBLT NEWBM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE))

```

**(ACESCALEREGION**

(\* MD "19-Jun-85 14:29")

```

[LAMBDA (TYPE)
  (PROG (TEMPREGION TEMPBM NEWBM)
    (ACE.TELLEM "Select a Region inside the Sequence" T)
    (SETQ TEMPREGION (GETREGION NIL NIL NIL (QUOTE ACE.EDIT.MOVE.REGION.ASST)
      NIL))
    (SETQ TEMPBM (BITMAPCREATE (fetch (REGION WIDTH) of TEMPREGION)
      (fetch (REGION HEIGHT) of TEMPREGION)
      1))
    (BITBLT ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      TEMPBM 0 0 NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE))
    [SETQ NEWBM (COND
      [(EQ TYPE (QUOTE SWEEP))
        (ACE.SCALE.BITMAP TEMPBM (ACEGETREGIONFACTOR (fetch (REGION WIDTH) of TEMPREGION)
          (fetch (REGION HEIGHT) of TEMPREGION))
        ]
      [(EQ TYPE (QUOTE XY))
        (ACE.SCALE.BITMAP TEMPBM (CONS (RNUMBER "%% scale in x")
          (RNUMBER "%% scale in y"))
        ]
      [(EQ TYPE (QUOTE X))
        (ACE.SCALE.BITMAP TEMPBM (CONS (RNUMBER "%% scale in x")
          100))
        ]
      [(EQ TYPE (QUOTE Y))
        (ACE.SCALE.BITMAP TEMPBM (CONS 100 (RNUMBER "%% scale in y"))
        ]
    ]
    (BITBLT TEMPBM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      NIL NIL (QUOTE INPUT)
      (QUOTE ERASE))
    (BITBLT NEWBM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE))

```

)

(\* \* LINEART FNS)

(DEFINEQ

**(ACE.EDIT.LINEART**

(\* MD "19-Jun-85 17:23")

[LAMBDA (ABITMAP)

(\* Entry into Lineart on ABITMAP.  
LSHIFT brings up options menu)

```
(RESETFORM (CURSOR ACE.EDIT.LINEART.SQUARE.CURSOR)
  (PROG (POINT&CODE DEVICE BRUSH MODE)
    (DECLARE (SPECVARS BRUSH MODE))
    (SETQ DEVICE (ACE.UPD.CONTROL.WINDOW (QUOTE DEVICE)))
    [COND
      ((EQ DEVICE (QUOTE MOUSE))
        (ACE.TELLEM "LEFT button Draws (puts Vertices)" T)
        (ACE.TELLEM "MIDDLE Stops drawing; IDLES (Square cursor)" (QUOTE L)))
      ((EQ DEVICE (QUOTE MM1201))
        (ACE.TELLEM "STYLUS Draws (puts down Vertices)" T)
        (ACE.TELLEM "BARREL Stops; IDLES (Square cursor)" (QUOTE L])
        (ACE.TELLEM "Right button selects drawing options" (QUOTE L))
        (ACE.TELLEM "To exit, use right button, or stylus + barrel (tablet)" (QUOTE L))
      ]
    LOOP
      (SETQ POINT&CODE (ACE.EDIT.POINT&CODE DEVICE))
      (COND
        ((EQUAL (CAR POINT&CODE)
          (QUOTE VERTEX))
          (ACE.EDIT.LINEART.DRAW ABITMAP POINT&CODE DEVICE BRUSH MODE))
        ((EQUAL (CAR POINT&CODE)
          (QUOTE OPTIONS))
          (IF (NOT (ACE.EDIT.LINEART.ADJ))
            THEN (RETURN NIL)))
        ((KEYDOWNP (QUOTE RSHIFT))
          (RETURN NIL))
        (forDuration 30 timerUnits (QUOTE MILLISECONDS))
        (GO LOOP])
```

(\* So ...LINEART.ADJ sees them.  
Overkill but a good reminder)

**(ACE.EDIT.LINEART.DRAW**

[LAMBDA (ABITMAP POINT&CODE DEVICE BRUSH MODE)

(\* PmT "24-Apr-85 15:02")

(\* The actual mule for LineArt. Draws the lines on ABITMAP and in the Sequence Window)

```
(PROG ((LASTVERTEX NIL)
  (LASTPOINT NIL)
  STUPID.DSP.FOR.ABITMAP TEMP.CURSOR)
  (SETQ STUPID.DSP.FOR.ABITMAP (DSPCREATE ABITMAP))
  (SETQ TEMP.CURSOR (CURSOR CROSSHAIRS))
  LOOP
    (AND LASTPOINT (ACE.EDIT.LINEART.TRACKLINE LASTVERTEX LASTPOINT BRUSH))
    (COND
      ((EQUAL (CAR POINT&CODE)
        (QUOTE VERTEX))
        (AND LASTVERTEX (DRAWLINE (ACE.MT.SEQX.AWX (CADR POINT&CODE))
          (ACE.MT.SEQY.AWY (CDDR POINT&CODE))
          (ACE.MT.SEQX.AWX (CAR LASTVERTEX))
          (ACE.MT.SEQY.AWY (CDR LASTVERTEX))
          BRUSH MODE ACE.SEQ.WINDOW)
        (DRAWLINE (CADR POINT&CODE)
          (CDDR POINT&CODE)
          (CAR LASTVERTEX)
          (CDR LASTVERTEX)
          BRUSH MODE STUPID.DSP.FOR.ABITMAP))
        (SETQ LASTVERTEX (CDR POINT&CODE))
        (SETQ LASTPOINT NIL))
      ((EQUAL (CAR POINT&CODE)
        (QUOTE TOGGLE))
        (CURSOR TEMP.CURSOR)
        (RETURN NIL))
      ((KEYDOWNP (QUOTE RSHIFT))
        (CURSOR TEMP.CURSOR)
        (RETURN NIL))
      (T (SETQ LASTPOINT (CDR POINT&CODE))
        (ACE.EDIT.LINEART.TRACKLINE LASTVERTEX LASTPOINT BRUSH)))
    (SETQ POINT&CODE (ACE.EDIT.POINT&CODE DEVICE))
    (GO LOOP])
```

**(ACE.EDIT.LINEART.ADJ**

[LAMBDA NIL

(\* MD "19-Jun-85 17:26")  
(\* Options for changing how lineart draws;  
affects brush and drawing mode)

```
(SELECTQ [MENU (OR ACE.EDIT.LINEART.ADJ.MENU (SETQ ACE.EDIT.LINEART.ADJ.MENU
  (create MENU
    ITEMS _ (QUOTE (("Replace Mode" (QUOTE REPLACE)
      "Puts bits there")
      ("Paint Mode" (QUOTE PAINT)
      "Cause lines to OR with
      background")
      ("Invert Mode" (QUOTE INVERT)
      "Lines will XOR with
      background")
      ("Erase Mode" (QUOTE ERASE)
```

```

"Erases whats there")
(" " NIL "Ah! you found the magic
  command:  nothing")
("Line width 1" 1 "Sets the Brush width
  to 1 pixel")
("Line width 2" 2 "Brush width 2
  pixels")
("Line width 3" 3 "Brush width 3
  pixels")
("Line width 4" 4 "Brush width 4
  pixels")
("Line width 5" 5 "Brush width 5
  pixels")
("Quit" (QUOTE QUIT)
  "Returns to Edit menu"))
CENTERFLG _ T
TITLE _ "Drawing Adjustments"]

(NIL NIL)
(PAINT (SETQ MODE (QUOTE PAINT)))
(INVERT (SETQ MODE (QUOTE INVERT)))
(REPLACE (SETQ MODE (QUOTE REPLACE)))
(ERASE (SETQ MODE (QUOTE ERASE)))
(1 (SETQ BRUSH 1))
(2 (SETQ BRUSH 2))
(3 (SETQ BRUSH 3))
(4 (SETQ BRUSH 4))
(5 (SETQ BRUSH 5))
(QUIT NIL)
NIL])

```

**(ACE.EDIT.LINEART.TRACKLINE**

[LAMBDA (VERTEX POINT BRUSH)

(\* PmT "24-Apr-85 15:25")

(\* Makes a line from the current vertex to the cursor; shows where a line would go and what it looks like)

```

(AND VERTEX POINT (DRAWLINE (ACE.MT.SEQX.AWX (CAR VERTEX))
  (ACE.MT.SEQY.AWX (CDR VERTEX))
  (ACE.MT.SEQX.AWX (CAR POINT))
  (ACE.MT.SEQY.AWX (CDR POINT))
  BRUSH
  (QUOTE INVERT)
  ACE.SEQ.WINDOW])
)

```

(\* \* OTHER EDITING STUFF)

(DEFINEQ

**(ACE.EDIT.MOVE.REGION**

[LAMBDA (ABITMAP)

(\* MD "21-Jun-85 13:52")

(\* Moves a region inside the sequence; similar to COMBINE.REGION except the old image may be erased (thus MOVE); region is confined to sequence by ACE.EDIT.MOVE.REGION.ASST)

```

(PROG (TEMP.REGION TEMP.BM)
  (ACE.TELLEM "Select a region inside the sequence" T)
  (SETQ TEMP.REGION (GETREGION NIL NIL NIL (QUOTE ACE.EDIT.MOVE.REGION.ASST)
    NIL))
  (SETQ TEMP.BM (BITMAPCREATE (fetch (REGION WIDTH) of TEMP.REGION)
    (fetch (REGION HEIGHT) of TEMP.REGION)
    1))
  (BITBLT ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMP.REGION))
    (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMP.REGION))
    TEMP.BM 0 0 NIL NIL (QUOTE INPUT)
    (QUOTE REPLACE))
  (SELECTQ [MENU (OR ACE.EDIT.MOVE.MENU (SETQ ACE.EDIT.MOVE.MENU
    (create MENU
      ITEMS _ (QUOTE ((Erase (QUOTE ERASE)
        "NAND the old image")
        (Invert (QUOTE XOR)
        "XOR the old image with itself")
        (Nothing NIL "Just leave the old image
        as is"))))
    CENTERFLG _ T
    TITLE _ "Do What with Old Image?"
    CHANGEOFFSETFLG _ T]
  (NIL NIL)
  (ERASE (BITBLT TEMP.BM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMP.REGION))
    (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMP.REGION))
    NIL NIL (QUOTE INPUT)
    (QUOTE ERASE)))
  (XOR (BITBLT TEMP.BM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMP.REGION))
    (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMP.REGION))
    NIL NIL (QUOTE INPUT)

```

```

                (QUOTE INVERT)))
      (NIL)
      (ACE.EDIT.PUTDOWN.BITMAP TEMP.BM ABITMAP))

```

**(ACE.EDIT.COMBINE.REGION**

[LAMBDA (ABITMAP)

(\* PmT "25-Apr-85 15:22")

(\* Grabs a region from the screen; makes it into a bitmap, then pastes it into the current frame;  
see ACE.EDIT.PUTDOWN.BITMAP)

```

(PROG (TEMP.BM TEMP.REGION)
  (ACE.TELLEM "Select a Region from Screen" T)
  (SETQ TEMP.REGION (GETREGION))
  (SETQ TEMP.BM (BITMAPCREATE (fetch (REGION WIDTH) of TEMP.REGION)
                              (fetch (REGION HEIGHT) of TEMP.REGION)
                              1))
  (BITBLT (SCREENBITMAP)
    (fetch (REGION LEFT) of TEMP.REGION)
    (fetch (REGION BOTTOM) of TEMP.REGION)
    TEMP.BM 0 0 NIL NIL (QUOTE INPUT)
    (QUOTE REPLACE))
  (ACE.EDIT.PUTDOWN.BITMAP TEMP.BM ABITMAP))

```

**(ACE.EDIT.TEXT**

[LAMBDA (ABITMAP)

(\* PmT "25-Apr-85 17:57")

(\* Puts text into current frame; actually, puts text into ABITMAP

and the sequence window)

```

(PROG ((FONT.FAMILY NIL)
      (FONT.SIZE NIL)
      (FONT.FACE NIL)
      (DUMB.DSP.FOR.BM (DSPCREATE ABITMAP))
      BMX BMY USERSTRING POSITION POINT&CODE FONT)
  (GETMOUSESTATE)
  (SETQ POSITION (CONS LASTMOUSEX LASTMOUSEY))
  (SETQ FONT.FAMILY (MENU (OR ACE.EDIT.TEXT.FONT.MENU
                              (create MENU
                                ITEMS _ [QUOTE ((Classic (QUOTE CLASSIC))
                                                  (Cream (QUOTE CREAM))
                                                  (Gacha (QUOTE GACHA))
                                                  (Helvetica (QUOTE
                                                                HELVETICA
                                                                ))
                                                  (Modern (QUOTE MODERN))
                                                  (TimesRoman (QUOTE
                                                                TIMESROMAN
                                                                ))
                                                  (" " NIL)
                                                  ("* Same *" (QUOTE SAME)
                                                                "Use same
                                                                descriptor as last
                                                                time")
                                                  ("* Other *" (QUOTE OTHER)
                                                                TITLE _ "Select a Font Family"
                                                                CENTERFLG _ T)))
                              POSITION)))
  (COND
    [(EQ FONT.FAMILY (QUOTE SAME))
     (SETQ FONT.FAMILY (FONTPROP (DSPFONT NIL ACE.SEQ.WINDOW)
                                (QUOTE FAMILY)))
     (SETQ FONT.SIZE (FONTPROP (DSPFONT NIL ACE.SEQ.WINDOW)
                               (QUOTE SIZE)))
     (SETQ FONT.FACE (FONTPROP (DSPFONT NIL ACE.SEQ.WINDOW)
                               (QUOTE FACE))
     (EQ FONT.FAMILY (QUOTE OTHER))
     (SETQ FONT.FAMILY (ACE.ASKEM "Enter a Font Family: " T)))
    (NULL FONT.FAMILY)
    (RETURN NIL)))
  (COND
    (FONT.SIZE NIL)
    [(NEQ (QUOTE OTHER)
      (SETQ FONT.SIZE (MENU (OR ACE.EDIT.TEXT.SIZE.MENU
                              (create MENU
                                ITEMS _
                                  [QUOTE (6 8 10 12 14 16 18 24 36 ("* Other *" (QUOTE OTHER)
                                                                TITLE _ "Select Font Size"
                                                                CENTERFLG _ T
                                                                MENUROWS _ 2)))
                              POSITION]
      (T (SETQ FONT.SIZE (ACE.ASKEM "Enter a Font Size: " T)
    (COND
      (FONT.FACE NIL)
      [(NEQ (QUOTE OTHER)
        (SETQ FONT.FACE
          (MENU (OR ACE.EDIT.TEXT.FACE.MENU

```

```

      (SETQ ACE.EDIT.TEXT.FACE.MENU
        (create MENU
          ITEMS _ (QUOTE (("Standard" (QUOTE STANDARD)
                                     "Normal font face")
                           ("Italic" (QUOTE ITALIC)
                                     "A standard face in italic")
                           ("Expanded" (QUOTE MRE))
                           ("Bold" (QUOTE BOLD)
                                     "A heavy bold face")
                           ("Bold Italic" (QUOTE BIR))
                           ("Compressed" (QUOTE MRC))
                           (" " NIL)
                           ("* Other *" (QUOTE OTHER))
                           (" " NIL)))
          TITLE _ "Select a Font Face"
          CENTERFLG _ T
          MENUROWS _ 3
          MENCOLUMNS _ 3)))

      POSITION]
      (T (SETQ FONT.FACE (ACE.ASKEM "Enter a Font Face: " T)
        (COND
          [(SETQ FONT (FONTCREATE FONT.FAMILY FONT.SIZE FONT.FACE NIL NIL T))
            (ACE.TELLEM "Ready to start Entering Text." T)
            (ACE.TELLEM "Position Mouse and Click LEFT." (QUOTE L))
            (until (EQ [CAR (SETQ POINT&CODE (ACE.EDIT.POINT&CODE (QUOTE MOUSE)
              (QUOTE VERTEX))
                do)
              (ACE.TELLEM "A RETURN Stops Text." T)
              (SETQ BMX (CADR POINT&CODE))
              (SETQ BMY (CDDR POINT&CODE))
              (MOVETO (ACE.MT.SEQX.AWX BMX)
                (ACE.MT.SEQY.AWY BMY)
                ACE.SEQ.WINDOW)
              (MOVETO BMX BMY DUMB.DSP.FOR.BM)
              [RESETFORM (DSPFONT FONT ACE.SEQ.WINDOW)
                (SETQ USERSTRING (PROMPTFORWORD NIL NIL NIL ACE.SEQ.WINDOW NIL NIL (CHARCODE (EOL ESCAPE LF
              ]
              (RESETFORM (DSPFONT FONT DUMB.DSP.FOR.BM)
                (AND USERSTRING (PRIN1 USERSTRING DUMB.DSP.FOR.BM)
              (T (ACE.TELLEM "Can't find any such Font. Aborted." T))

```

**(ACE.EDIT.TEXTURE.REGION**

[LAMBDA (ABITMAP SHADE)

(\* PmT "25-Apr-85 15:33")

(\* Textures a region in ABITMAP; SHADE is optional; if not given, user is asked for a shade)

```

      (PROG (OPERATION SHADEREGION)
        [COND
          (SHADE NIL)
          [(NEQ (QUOTE OTHER)
            (SETQ SHADE (MENU (OR ACE.EDIT.TEXTURE.MENU (SETQ ACE.EDIT.TEXTURE.MENU (
              ACE.EDIT.CREATE.MENU.TEXTURES
            ]
            (T (SETQ SHADE (EDITSHADE)
              (OR SHADE (RETURN NIL))
              (ACE.TELLEM "Select a Region in the Sequence" T)
              (SETQ SHADEREGION (GETREGION))
              [SETQ OPERATION (COND
                ((EQ SHADE WHITESHADE)
                  (QUOTE REPLACE))
                ((MENU (OR ACE.EDIT.PUTDOWN.MENU (SETQ ACE.EDIT.PUTDOWN.MENU
                  (create MENU
                    ITEMS _ (QUOTE (("Paint" (QUOTE PAINT)
                                      "ORs with Frame")
                                      ("Replace" (QUOTE REPLACE)
                                      "Puts onto Frame")
                                      ("Invert" (QUOTE INVERT)
                                      "XORs with Frame")
                                      ("Erase" (QUOTE ERASE)
                                      "NANDs with Frame")
                                      ))
                    CENTERFLG _ T
                    TITLE _ "Select a Drawing Mode"]
                  (replace (REGION LEFT) of SHADEREGION with (ACE.MT.SCRX.AWX (fetch (REGION LEFT) of SHADEREGION)))
                  (replace (REGION BOTTOM) of SHADEREGION with (ACE.MT.SCRY.AWY (fetch (REGION BOTTOM) of SHADEREGION)))
                  (BITBLT NIL NIL NIL ACE.SEQ.WINDOW (fetch (REGION LEFT) of SHADEREGION)
                    (fetch (REGION BOTTOM) of SHADEREGION)
                    (fetch (REGION WIDTH) of SHADEREGION)
                    (fetch (REGION HEIGHT) of SHADEREGION)
                    (QUOTE TEXTURE)
                    OPERATION SHADE (ACE.MT.SEQ.AW.REGION))
                  (BITBLT NIL NIL NIL ABITMAP (ACE.MT.AWX.SEQX (fetch (REGION LEFT) of SHADEREGION)
                    (ACE.MT.AWY.SEQY (fetch (REGION BOTTOM) of SHADEREGION)
                    (fetch (REGION WIDTH) of SHADEREGION)
                    (fetch (REGION HEIGHT) of SHADEREGION)
                    (QUOTE TEXTURE)

```

OPERATION SHADE]])

**(ACE.EDIT.TEXTURE.AREA**

[LAMBDA (ABITMAP SHADE)

(\* PmT "2-May-85 19:47")

(\* Does an area flood on a bounded region. ABITMAP is the frame (required); SHADE is an optional shade arg (SMALLP); Works by making a bitmap copy of a user selected region. Then flooding whatever area is enclosed with a user seed point.)

```
(PROG (BOUNDING.REGION SEED.POINT TEMP.BM)
  (ACE.TELLEM "Select a Maximum Bounding Region." T)
  (SETQ BOUNDING.REGION (GETREGION NIL NIL NIL (QUOTE ACE.EDIT.MOVE.REGION.ASST)
    NIL))
  (ACE.TELLEM "Select a Starting Point Inside the Region." T)
  (SETQ SEED.POINT (GETPOSITION))
  (ACE.TELLEM "Select a Fill Shade." T)
  [COND
    (SHADE)
    [(NEQ (QUOTE OTHER)
      (SETQ SHADE (MENU (OR ACE.EDIT.TEXTURE.MENU (SETQ ACE.EDIT.TEXTURE.MENU (
        ACE.EDIT.CREATE.MENU.TEXTURES
      ]
      (T (SETQ SHADE (EDITSHADE]
    [COND
      ((NOT (INSIDEP BOUNDING.REGION SEED.POINT))
        (ACE.TELLEM "Your Seed Point lies Outside of your Region." T)
        (ACE.TELLEM "Texture Area Fill Aborted." (QUOTE L))
        (RETURN NIL))
      ((NULL SHADE)
        (ACE.TELLEM "No Shade selected. Aborted." T)
        (RETURN NIL))
      (T (SETQ SEED.POINT (CONS (IDIFFERENCE (CAR SEED.POINT)
        (fetch (REGION LEFT) of BOUNDING.REGION))
        (IDIFFERENCE (CDR SEED.POINT)
        (fetch (REGION BOTTOM) of BOUNDING.REGION]
        (SETQ TEMP.BM (BITMAPCREATE (fetch (REGION WIDTH) of BOUNDING.REGION)
        (fetch (REGION HEIGHT) of BOUNDING.REGION)
        1))
        (BITBLT ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of BOUNDING.REGION))
        (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of BOUNDING.REGION))
        TEMP.BM 0 0 NIL NIL (QUOTE INPUT)
        (QUOTE REPLACE))
        (ACE.TELLEM "Texturing Area..." T)
        [RESETFORM (CURSOR WAITINGCURSOR)
          (ACE.FILLWITHTEXTURE TEMP.BM SHADE (CAR SEED.POINT)
            (CDR SEED.POINT)
            (LOGXOR 1 (BITMAPBIT TEMP.BM (CAR SEED.POINT)
              (CDR SEED.POINT]
          (BITBLT TEMP.BM 0 0 ACE.SEQ.WINDOW (ACE.MT.SCRX.AWX (fetch (REGION LEFT) of BOUNDING.REGION))
            (ACE.MT.SCRY.AWY (fetch (REGION BOTTOM) of BOUNDING.REGION))
            NIL NIL (QUOTE INPUT)
            (QUOTE REPLACE)
            NIL NIL)
        (COND
          ((ACE.CONFIRMIT "Click LEFT to Confirm Fill as Shown." T)
            (BITBLT TEMP.BM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of BOUNDING.REGION))
              (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of BOUNDING.REGION))
              NIL NIL (QUOTE INPUT)
              (QUOTE REPLACE)
              NIL NIL))
          (T (ACE.TELLEM "Texture Area Fill Aborted." T]))
```

**(ACE.EDIT.PAINT**

[LAMBDA (ABITMAP)

(\* MJD "22-May-86 17:18")

(\* Hacked from rrb "18-OCT-83 18:37")

(\* Paint on current frame with either Mouse or Tablet)

(\* should make sure cursor has moved or a button has change before proceeding with the inner loop.)

(\* has some of the stuff to allow the brush to be an arbitrary bitmap but not all.)

```
(ACE.TELLEM "Left (Mouse), Stylus (Tablet) Paints." T)
(ACE.TELLEM "Middle (Mouse), Barrel (Tablet) Erases." (QUOTE L))
(ACE.TELLEM "Right (Mouse) or Left Shift for Menu." (QUOTE L))
(ACE.TELLEM "To Quit, select Quit from Menu." (QUOTE L))
(BITBLT ABITMAP 0 0 ACE.SEQ.WINDOW ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF NIL NIL (QUOTE INPUT)
  (QUOTE REPLACE)
  NIL NIL)
(PROG (OLDX OLDY (NEWBRUSHQ T)
  BRUSH WINDOW (ORIG.CURSOR (CURSOR))
  (ORIG.REGION (DSPCLIPPINGREGION NIL ACE.SEQ.WINDOW)))
  (SETQ WINDOW ACE.SEQ.WINDOW)
  (SETQ WINDOW (\INSUREWINDOW WINDOW))
  [RESETLST
    (RESETSAVE NIL (LIST (QUOTE CURSOR)
```



```

(CURSOR)))
(PROG (DS HOTX HOTY)
  (TOTOPW WINDOW)
  (* look for a previously stored brush.)
  [COND
    ((SETQ BRUSH (WINDOWPROP WINDOW (QUOTE PAINTBRUSH)))
     (SETQ PAINTCOMMANDMODE (CAR BRUSH))
     (SETQ PAINTCOMMANDSHADE (CADR BRUSH))
     (SETQ PAINTCOMMANDBRUSH (CADDR BRUSH))
     (SETQ DS (WINDOWPROP WINDOW (QUOTE DSP)))
    BRUSHLP
    [AND NEWBRUSHQ (SETQ BRUSH (COND
      ((BITMAP PAINTCOMMANDBRUSH))
      ((EQ (CAR PAINTCOMMANDBRUSH)
        (QUOTE other))
        (SETQ NEWBRUSHQ NIL)
        (ACE.EXTRACTBM))
      (T (\GETBRUSH PAINTCOMMANDBRUSH))
    ]
    [AND (BITMAP PAINTCOMMANDBRUSH)
      (SETQ PAINTCOMMANDBRUSH (QUOTE (SQUARE 2]
      (* clear cursor)
    (BITBLT NIL NIL NIL (CURSORBITMAP)
      0 0 CURSORWIDTH CURSORHEIGHT (QUOTE TEXTURE)
      (QUOTE REPLACE)
      WHITESHADE)
      (* put lower left part of brush shape in cursor)
      (* BITBLT BRUSH 0 0 (SCREENBITMAP) 0 0 NIL NIL
      (QUOTE INPUT) (QUOTE REPLACE))
      (* set the hot spot to the middle of the brush.)
    [CURSORHOTSPOT (create POSITION
      XCOORD _ (SETQ HOTX (IDIFFERENCE (IMIN (fetch BITMAPWIDTH of BRUSH)
        CURSORWIDTH)
        2))
      YCOORD _ (SETQ HOTY (IDIFFERENCE (IMIN (fetch BITMAPHEIGHT of BRUSH)
        CURSORHEIGHT)
        2]
    PAINTLP
    (ACE.GET.DEVICE.STATE&CURSOR)
    [COND
      ((KEYDOWNP (QUOTE RSHIFT))
       (RETURN))
      ((OR (LASTMOUSESTATE RIGHT)
        (KEYDOWNP (QUOTE LSHIFT)))
       (COND
         ((OR (INSIDE? (DSPCLIPPINGREGION NIL DS)
           (LASTMOUSEX DS)
           (LASTMOUSEY DS))
          (NOT (WHICHW LASTMOUSEX LASTMOUSEY)))
          (* inside the interior, give command menu)
          (SELECTQ [MENU (COND
            ((type? MENU PAINTCOMMANDMENU)
             PAINTCOMMANDMENU)
            (T (SETQ PAINTCOMMANDMENU (create MENU
              ITEMS _
              (QUOTE ((HardCopy (QUOTE HARDCOPY
                )
                "Makes a press
                file of the window
                and prints it")
                (SetMode (QUOTE MODE)
                "Allows
                specification of
                how new bits are
                merged")
                (SetShade (QUOTE SHADE)
                "Allows
                specification of
                new shade.")
                (SetShape (QUOTE SHAPE)
                "Allows
                specification of
                brush shape")
                (SetSize (QUOTE SIZE)
                "Allows
                specification of
                the brush size")
                (QUIT (QUOTE QUIT)
                "Exits painting
                mode"))
              CHANGEOFFSETFLG _ T]
            (SHADE (SETQ PAINTCOMMANDSHADE (OR (PAINTW.READBRUSHSHADE)
              PAINTCOMMANDSHADE))
              (GO BRUSHLP))
            (MODE (SETQ PAINTCOMMANDMODE (OR (PAINTW.READMODE)
              PAINTCOMMANDMODE))
              (GO BRUSHLP))
            (SHAPE (RPLACA PAINTCOMMANDBRUSH (OR (ACE.READBRUSHSHAPE)
              (CAR PAINTCOMMANDBRUSH)))
              (SETQ NEWBRUSHQ T)

```

```

        (GO BRUSHLP))
      (SIZE (RPLACA (CDR PAINTCOMMANDBRUSH)
        (OR (PAINTW.READBRUSHSIZE)
          (CADR PAINTCOMMANDBRUSH)))
        (GO BRUSHLP))
      (QUIT (WINDOWPROP ACE.SEQ.WINDOW (QUOTE PAINTBRUSH)
        (LIST PAINTCOMMANDBRUSH PAINTCOMMANDBRUSH)))
        (SETQ PAINTCOMMANDBRUSH (QUOTE (ROUND 16)))
        (RETURN))
      (HARDCOPY (RESETFORM (TTYDISPLAYSTREAM PROMPTWINDOW)
        (HARDCOPYW WINDOW)))
      NIL))
    (T
      NIL)))
  [ (AND (LASTMOUSESTATE LEFT)
    (OR (EQ PAINTCOMMANDBRUSH (QUOTE REPLACE))
      (NEQ PAINTCOMMANDBRUSH BLACKSHADE)))
    (* painting in grey is slightly harder.)
  (COND
    ((EQ PAINTCOMMANDBRUSH (QUOTE REPLACE))
      (* erase what is there now)
      (BITBLT BRUSH 0 0 DS (IDIFFERENCE (LASTMOUSEX DS)
        HOTX)
        (IDIFFERENCE (LASTMOUSEY DS)
          HOTY)
        NIL NIL (QUOTE INPUT)
        (QUOTE ERASE))
      (* put in grey)
      (BITBLT BRUSH 0 0 DS (IDIFFERENCE (LASTMOUSEX DS)
        HOTX)
        (IDIFFERENCE (LASTMOUSEY DS)
          HOTY)
        NIL NIL (QUOTE MERGE)
        (QUOTE PAINT)
        PAINTCOMMANDBRUSH))
      (T (BITBLT BRUSH 0 0 DS (IDIFFERENCE (LASTMOUSEX DS)
        HOTX)
        (IDIFFERENCE (LASTMOUSEY DS)
          HOTY)
        NIL NIL (QUOTE MERGE)
        PAINTCOMMANDBRUSH PAINTCOMMANDBRUSH))
      (LASTMOUSESTATE (OR MIDDLE LEFT))
      (BITBLT BRUSH 0 0 DS (IDIFFERENCE (LASTMOUSEX DS)
        HOTX)
        (IDIFFERENCE (LASTMOUSEY DS)
          HOTY)
        NIL NIL (QUOTE INPUT)
        (COND
          ((LASTMOUSESTATE MIDDLE)
            (QUOTE ERASE))
          (T PAINTCOMMANDBRUSH))
      ((LASTMOUSESTATE UP)
        (* Idle -
          just mark brush loc.)
        (BITBLT BRUSH 0 0 DS (SETQ OLDX (IDIFFERENCE (LASTMOUSEX DS)
          HOTX))
          (SETQ OLDY (IDIFFERENCE (LASTMOUSEY DS)
            HOTY))
          NIL NIL (QUOTE INPUT)
          (QUOTE INVERT))
        (DISMISS 2)
        (BITBLT BRUSH 0 0 DS OLDX OLDY NIL NIL (QUOTE INPUT)
          (QUOTE INVERT))
        (GO PAINTLP))
      (WINDOWPROP WINDOW (QUOTE PAINTBRUSH)
        (LIST PAINTCOMMANDBRUSH PAINTCOMMANDBRUSH (COPY PAINTCOMMANDBRUSH)))
      (BITBLT WINDOW ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF ABITMAP 0 0 ACE.SEQ.WIDTH ACE.SEQ.HEIGHT
        (QUOTE INPUT)
        (QUOTE REPLACE)
        NIL NIL))
      (WINDOWPROP ACE.SEQ.WINDOW (QUOTE PAINTBRUSH)
        (LIST PAINTCOMMANDBRUSH PAINTCOMMANDBRUSH (OR (BITMAPP BRUSH)
          PAINTCOMMANDBRUSH]))

```

**(ACE.FILLWITHTEXTURE**

[LAMBDA (BitmapOrWindow Texture X Y BoundaryValue)

(\* PmT " 2-May-85 18:45")  
(\* hdj " 5-Mar-85 15:53")

(\* This code created by Herb Jellinek and Kelly Roach. Renamed for organizational purposes.  
Paints in TEXTURE into supplied BitmapOrWindow (Only a bitmap here);  
BoundaryValue should be either a 1 or 0 (No color bitmaps or anything fancy for now);  
X and Y specify the seed point for the fill)

```

(PROG ((COPYBM (COPYALL BitmapOrWindow)))
  (ACE.SCANLINESEEDFILL COPYBM X Y BoundaryValue BoundaryValue)
  (BITBLT BitmapOrWindow 0 0 COPYBM 0 0 NIL NIL (QUOTE INPUT)
    (QUOTE INVERT)))

```

```

(BITBLT COPYBM 0 0 COPYBM 0 0 NIL NIL (QUOTE MERGE)
 (QUOTE TEXTURE)
 Texture)
(BITBLT COPYBM 0 0 BitmapOrWindow 0 0 NIL NIL (QUOTE INPUT)
 (QUOTE PAINT))
BitmapOrWindow])

```

**(ACE.SCANLINESEEDFILL**

```

[LAMBDA (BitmapOrWindow X Y BoundaryValue FillValue)
  (* PmT "2-May-85 20:19")
  (* hdj "30-Jan-85 15:01")
  (* This code created by Herb Jellinek and Kelly Roach.
  Renamed for organizational purposes)

  (PROG (Xcoord Ycoord STACK SaveX SaveY XLeft XRight XMax YMax)
    [if (BITMAPP BitmapOrWindow)
      then (SETQ XMax (SUB1 (BITMAPWIDTH BitmapOrWindow)))
            (SETQ YMax (SUB1 (BITMAPHEIGHT BitmapOrWindow)))
      else [SETQ XMax (SUB1 (WINDOWPROP BitmapOrWindow (QUOTE WIDTH)
            (SETQ YMax (SUB1 (WINDOWPROP BitmapOrWindow (QUOTE HEIGHT)
              (* "initialize stack")

  (ACE.PUSHPOS X Y STACK)
  (while STACK do
    (* get seed pixel and set to new value)
    (ACE.POPPOS STACK SaveX SaveY)
    (BITMAPBIT BitmapOrWindow SaveX SaveY FillValue)
    (* fill span to right of seed pixel)
    (SETQ XRight XMax)
    (for Xcoord from (ADD1 SaveX) while (ILEQ Xcoord XMax)
      do (if (NEQ (BITMAPBIT BitmapOrWindow Xcoord SaveY)
        BoundaryValue)
        then (BITMAPBIT BitmapOrWindow Xcoord SaveY FillValue)
        else (* save the extreme right pixel)
          (SETQ XRight (SUB1 Xcoord))
          (RETURN)))
        (* fill span to left of seed pixel)
    (SETQ XLeft 0)
    (for Xcoord from (SUB1 SaveX) by -1 while (IGEQ Xcoord 0)
      do (if (NEQ (BITMAPBIT BitmapOrWindow Xcoord SaveY)
        BoundaryValue)
        then (BITMAPBIT BitmapOrWindow Xcoord SaveY FillValue)
        else (* save the extreme left pixel)
          (SETQ XLeft (ADD1 Xcoord))
          (RETURN)))
        (* Push seed points for scan line above.
        *)

    [COND
      ((ILESSP SaveY YMax)
        (SETQ Ycoord (ADD1 SaveY))
        (for Xcoord from XLeft to XRight
          when [AND (NEQ (BITMAPBIT BitmapOrWindow Xcoord Ycoord)
            BoundaryValue)
            (OR (EQ Xcoord XRight)
              (OR (EQ (BITMAPBIT BitmapOrWindow (ADD1 Xcoord)
                Ycoord)
                BoundaryValue)
                (EQ (BITMAPBIT BitmapOrWindow (ADD1 Xcoord)
                Ycoord)
                FillValue)]
            do (ACE.PUSHPOS Xcoord Ycoord STACK)
            (* Push seed points for scan line below.
            *)

    (COND
      ((IGREATERP SaveY 0)
        (SETQ Ycoord (SUB1 SaveY))
        (for Xcoord from XLeft to XRight
          when [AND (NEQ (BITMAPBIT BitmapOrWindow Xcoord Ycoord)
            BoundaryValue)
            (OR (EQ Xcoord XRight)
              (OR (EQ (BITMAPBIT BitmapOrWindow (ADD1 Xcoord)
                Ycoord)
                BoundaryValue)
                (EQ (BITMAPBIT BitmapOrWindow (ADD1 Xcoord)
                Ycoord)
                FillValue)]
            do (ACE.PUSHPOS Xcoord Ycoord STACK])

```

**(ACE.EDIT.CREATE.MENU.TEXTURES**

```

[LAMBDA NIL
  (* PmT "25-Apr-85 15:27")
  (* Creates a textures menu for TEXTURE.REGION and FILL
  routines)

  (PROG (TEMP.BM)
    (RETURN (create MENU
      ITEMS _
      (NCONC1 (for TEXTURE
        in (QUOTE (65535 34850 43605 63624 42405 4080 64250 26214 65488 34925 15 4680
          33825 1185 1 3784 3591))
        collect (PROGN (BITBLT NIL NIL NIL (SETQ TEMP.BM (BITMAPCREATE 36 36 1))
          4 4 28 28 (QUOTE TEXTURE)
          (QUOTE REPLACE)

```

```

TEXTURE NIL)
  (LIST TEMP.BM TEXTURE)))
(QUOTE ("* Other *" (QUOTE OTHER)
  "Make your own shade")))
TITLE _ "Texture Menu"
CENTERFLG _ T
MENUROWS _ 3])

```

**(ACE.EDIT.PUTDOWN.BITMAP**

[LAMBDA (IMAGE.BM TARGET.BM)

(\* MD "21-Jun-85 13:56")

(\* Pastes IMAGE.BM onto TARGET.BM; IMAGE.BM is tied to mouse until left click;  
asks user how to combine the two)

```

(PROG (MODE POINT)
  (ACE.TELLEM "Click Left to paste down image." T)
  (SETQ POINT (PROG (OLD.X OLD.Y)
    LOOP
      (BITBLT IMAGE.BM 0 0 ACE.SEQ.WINDOW (SETQ OLD.X (LASTMOUSEX ACE.SEQ.WINDOW))
        (SETQ OLD.Y (LASTMOUSEY ACE.SEQ.WINDOW))
        NIL NIL (QUOTE INPUT)
        (QUOTE INVERT)
        NIL
        (ACE.MT.SEQ.AW.REGION))
      (GETMOUSESTATE)
      (ACE.UPD.CONTROL.WINDOW (QUOTE CURSOR)
        (CONS (ACE.MT.SCRX.SEQX LASTMOUSEX)
          (ACE.MT.SCRY.SEQY LASTMOUSEY)))
      [COND
        [(LASTMOUSESTATE LEFT)
          (do (GETMOUSESTATE) until (LASTMOUSESTATE UP))
          (BITBLT IMAGE.BM 0 0 ACE.SEQ.WINDOW OLD.X OLD.Y NIL NIL (QUOTE INPUT)
            (QUOTE INVERT)
            NIL
            (ACE.MT.SEQ.AW.REGION))
          (RETURN (CONS (LASTMOUSEX ACE.SEQ.WINDOW)
            (LASTMOUSEY ACE.SEQ.WINDOW))
            (T (BITBLT IMAGE.BM 0 0 ACE.SEQ.WINDOW OLD.X OLD.Y NIL NIL (QUOTE INPUT)
              (QUOTE INVERT)
              NIL
              (ACE.MT.SEQ.AW.REGION]
            (GO LOOP)))
        ]
      [SETQ MODE (MENU (OR ACE.EDIT.PUTDOWN.MENU (SETQ ACE.EDIT.PUTDOWN.MENU
        (create MENU
          ITEMS _ (QUOTE (("Paint" (QUOTE PAINT)
            "ORs with Frame")
            ("Replace" (QUOTE REPLACE)
            "Puts onto Frame")
            ("Invert" (QUOTE INVERT)
            "XORs with Frame")
            ("Erase" (QUOTE ERASE)
            "NANDs with Frame"))))
          CENTERFLG _ T
          TITLE _ "Select a Drawing Mode"
          CHANGEOFFSETFLG _ T]
        (BITBLT IMAGE.BM 0 0 ACE.SEQ.WINDOW (CAR POINT)
          (CDR POINT)
          NIL NIL (QUOTE INPUT)
          MODE NIL (ACE.MT.SEQ.AW.REGION))
        (BITBLT IMAGE.BM 0 0 TARGET.BM (ACE.MT.AWX.SEQX (CAR POINT))
          (ACE.MT.AWY.SEQY (CDR POINT))
          NIL NIL (QUOTE INPUT)
          MODE NIL])

```

**(ACE.EDIT.MOVE.REGION.ASST**

[LAMBDA (FIXED MOVE EXTRA)

(\* PmT "23-Jan-85 20:03")

```

(COND
  [(NULL MOVE)
    (ACE.NEW.SEQ.ASST (CONS (ACE.MT.SCRX.SEQX (CAR FIXED))
      (ACE.MT.SCRY.SEQY (CDR FIXED)))
    MOVE EXTRA)
  (COND
    ((INSIDE (ACE.MT.SEQ.SCR.REGION)
      FIXED)
      FIXED)
    (T (ACE.RETURN.CLOSEST.VERTEX FIXED (ACE.MT.SEQ.SCR.REGION]
  (T (ACE.NEW.SEQ.ASST FIXED MOVE EXTRA)
    (COND
      ((INSIDE (ACE.MT.SEQ.SCR.REGION)
        MOVE)
        MOVE)
      (T (ACE.RETURN.CLOSEST.VERTEX MOVE (ACE.MT.SEQ.SCR.REGION]

```

**(ACEEDITBM**

```

[LAMBDA (TYPE) (* MJD "23-May-86 15:51")
  (PROG (TEMPREGION TEMPBM NEWBM)
    (ACE.TELLEM "Select a Region inside the Sequence" T)
    (SETQ TEMPREGION (GETREGION NIL NIL NIL (QUOTE ACE.EDIT.MOVE.REGION.ASST)
      NIL))
    (SETQ TEMPBM (BITMAPCREATE (fetch (REGION WIDTH) of TEMPREGION)
      (fetch (REGION HEIGHT) of TEMPREGION)
      1))
    (BITBLT ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      TEMPBM 0 0 NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE))
    (SETQ NEWBM (EDITBM TEMPBM))
    (BITBLT TEMPBM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      NIL NIL (QUOTE INPUT)
      (QUOTE ERASE))
    (BITBLT NEWBM 0 0 ABITMAP (ACE.MT.SCRX.SEQX (fetch (REGION LEFT) of TEMPREGION))
      (ACE.MT.SCRY.SEQY (fetch (REGION BOTTOM) of TEMPREGION))
      NIL NIL (QUOTE INPUT)
      (QUOTE REPLACE))
  )

```

**(ACE.READBRUSHSHAPE**

```

[LAMBDA NIL (* MJD "15-May-86 15:15")
  (MENU (create MENU
    ITEMS _ (QUOTE (DIAGONAL VERTICAL HORIZONTAL SQUARE ROUND other))
  )

```

(\* \* TABLET AND SUPPORT FNS)

```

(DEFINEQ

```

**(ACE.EDIT.POINT&CODE**

```

[LAMBDA (DEVICE)
  (* MD "19-Jun-85 16:56")
  (* THIS MESS RETURNS (CODE X NIL) FOR THE DEVICE
  SELECTED)
  (* CODE IS NIL, VERTEX OR TOGGLE)

```

(\* X AND Y ARE RELATIVE TO THE SEQUENCE. USUALLY THE SEQUENCE'S 0,0 IS AT THE WINDOW'S 0,0  
THEREFORE, COORS IN THE A.C.W ARE SEQUENCE COORS, NOT WINDOW COORS.  
E.S.L.DRAW TAKES CARE TO CORRECT FOR ANY DIFFERENCE)

```

(PROG (POINT&CODE XCOOR YCOOR)
  (ACE.GET.DEVICE.STATE DEVICE)
  (SETQ XCOOR (ACE.MT.SCRX.SEQX LASTMOUSEX))
  (SETQ YCOOR (ACE.MT.SCRY.SEQY LASTMOUSEY))
  [SETQ POINT&CODE (DECODEBUTTONS (QUOTE (LEFT MIDDLE RIGHT)
  [COND
    (POINT&CODE (FLIPCURSOR)
      (until (EQ LASTMOUSEBUTTONS 0) do (ACE.GET.DEVICE.STATE DEVICE))
      (FLIPCURSOR)
      (SETQ POINT&CODE (COND
        ((EQUAL POINT&CODE (QUOTE (LEFT)))
          (QUOTE VERTEX))
        ((EQUAL POINT&CODE (QUOTE (MIDDLE)))
          (QUOTE TOGGLE))
        ((EQUAL POINT&CODE (QUOTE (RIGHT)))
          (QUOTE OPTIONS))
      )
    )
  (COND
    ((EQ DEVICE (QUOTE MOUSE))
      NIL)
    ((EQ DEVICE (QUOTE MM1201))
      (\SETCURSORPOSITION LASTMOUSEX LASTMOUSEY)))
  (ACE.UPD.CONTROL.WINDOW (QUOTE CURSOR)
    (CONS XCOOR YCOOR))
  (RETURN (CONS POINT&CODE (CONS XCOOR YCOOR]))

```

**(ACE.GET.DEVICE.STATE**

```

[LAMBDA (DEVICE)
  (* PmT "24-Apr-85 16:57")
  (* Updates LASTMOUSEX LASTMOUSEY
  LASTMOUSEBUTTONS LASTKEYBOARD based on DEVICE
  (mouse or tablet (MM1201)))

```

```

(PROG (POINT&CODE)
  (COND
    ((EQ DEVICE (QUOTE MOUSE))
      (GETMOUSESTATE))
    ((EQ DEVICE (QUOTE MM1201))
      (SETQ POINT&CODE (ACE.MM1201POLL 1))
      (COND
        ((BITTEST (CDR POINT&CODE)
          64)
          NIL)
        (T (COND

```

```

      ((BITTEST (CDR POINT&CODE)
        2)
      (SETQ LASTMOUSEBUTTONS 1))
      ((BITTEST (CDR POINT&CODE)
        1)
      (SETQ LASTMOUSEBUTTONS 4))
      (T (SETQ LASTMOUSEBUTTONS 0)))
      (SETQ LASTMOUSEX (CAAR POINT&CODE))
      (SETQ LASTMOUSEY (CDAR POINT&CODE))
      (SETQ LASTKEYBOARD (\EVENTKEYS])

```

**(ACE.GET.DEVICE.STATE&CURSOR**

[LAMBDA NIL

(\* PmT "25-Apr-85 14:10")

(\* Updates LASTMOUSEX LASTMOUSEY LASTMOUSEBUTTONS LASTKEYBOARD based on the window prop value of DEVICE; Also, puts cursor info in status window (sequence referenced))

```

(PROG (DEVICE POINT&CODE)
  (SETQ DEVICE (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE ACE.DEVICE)))
  [COND
    ((EQ DEVICE (QUOTE MOUSE))
     (GETMOUSESTATE))
    ((EQ DEVICE (QUOTE MM1201))
     (SETQ POINT&CODE (ACE.MM1201POLL 1))
     (COND
       ((BITTEST (CDR POINT&CODE)
        64)
        NIL)
       (T (COND
          ((BITTEST (CDR POINT&CODE)
            2)
           (SETQ LASTMOUSEBUTTONS 1))
          ((BITTEST (CDR POINT&CODE)
            1)
           (SETQ LASTMOUSEBUTTONS 4))
          (T (SETQ LASTMOUSEBUTTONS 0)))
          (SETQ LASTMOUSEX (CAAR POINT&CODE))
          (SETQ LASTMOUSEY (CDAR POINT&CODE))
          (\SETCURSORPOSITION LASTMOUSEX LASTMOUSEY)
          (SETQ LASTKEYBOARD (\EVENTKEYS])
          (ACE.MAC.CW.INFO.CLIP (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
            (CONS (ACE.MT.SCRX.SEQX LASTMOUSEX)
              (ACE.MT.SCRY.SEQY LASTMOUSEY]))

```

**(ACE.EXTRACTBM**

[LAMBDA NIL

(\* MJD "22-May-86 15:16")

```

  (PROG (SHAPE BMNAME)
    [SETQ SHAPE (GETREGION 4 4 (QUOTE (32 . 32)
      (SETQ BMNAME (BITMAPCREATE (CADDR SHAPE)
        (CADDRD SHAPE)))
    (BITBLT (SCREENBITMAP)
      (CAR SHAPE)
      (CADR SHAPE)
      (EVAL BMNAME)
      0 0 (CADDR SHAPE)
      (CADDRD SHAPE)
      (QUOTE INPUT)
      (QUOTE REPLACE))
    (RETURN BMNAME)]

```

**(ACE.EDIT.REDRAW.ABITMAP**

[LAMBDA (ABITMAP)

(\* PmT "2-Jan-85 18:28")

```

  (BITBLT ABITMAP 0 0 ACE.SEQ.WINDOW ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF NIL NIL (QUOTE INPUT)
    (QUOTE REPLACE)
    NIL NIL))

```

**(ACE.SCALE.BITMAP**

[LAMBDA (BITMAP FACTOR)

(\* PmT "2-May-85 20:19")

(\* SCALES BITMAPS BY AN ARBITRARY AMOUNT OF 2 DECIMAL PLACES.  
 FACTOR CAN BE OF THE FOLLOWING FORMS: I (AN INTEGER REPRESENTING A PERCENTAGE AMOUNT;  
 E.G. I=67 MEANS REDUCE THE X AND Y AXIS TO 67% OF THEIR ORIGINAL);  
 R (A REAL; E.G. R=1.3 MEANS INCREASE THE X AND Y AXIS BY A FACTOR OF 1.3);  
 (IX . IY) (A DOTTED PAIR OF INTEGERS; E.G. (75 . 125) MEANS REDUCE THE X AXIS TO 75% OF ORIGINAL;  
 INCREASE Y TO 125% OF ORIGINAL); (RX . RY) (A DOTTED PAIR OF REALS;  
 E.G. (2.3 . 0.81) MEANS 2.3 TIMES ORIGINAL X AXIS, 0.81 TIMES ORIGINAL Y))

```

(PROG (XFACTOR YFACTOR DELTAX DELTAY XROUND YROUND BITMAPWIDTH BITMAPHEIGHT HEIGHT-1 RASTERWIDTH BITMAPBASE
  NEWBITMAP NEWHEIGHT-1 NEWBITMAPBASE NEWRASTERWIDTH ORIGBASE NEWBASE ORIGWORD NEWWORD XSTART
  YSTART ENDX ENDY ONLINE)
  (OR (type? BITMAP BITMAP)
    (\ILLEGAL.ARG BITMAP))
  (SETQ BITMAPWIDTH (fetch (BITMAP BITMAPWIDTH) of BITMAP))

```

```

(SETQ BITMAPHEIGHT (fetch (BITMAP BITMAPHEIGHT) of BITMAP))
(COND
  ((NUMBERP FACTOR)
    (SETQ XFACTOR FACTOR)
    (SETQ YFACTOR FACTOR)
    (POSITIONP FACTOR)
    (SETQ XFACTOR (CAR FACTOR))
    (SETQ YFACTOR (CDR FACTOR)))
  (T (\ILLEGAL.ARG FACTOR)))
[AND (FLOATP XFACTOR)
  (SETQ XFACTOR (FIX (FTIMES XFACTOR 100))
[AND (FLOATP YFACTOR)
  (SETQ YFACTOR (FIX (FTIMES YFACTOR 100))
(SETQ XFACTOR (IMIN SCREENWIDTH XFACTOR))
(SETQ YFACTOR (IMIN SCREENHEIGHT YFACTOR))
(COND
  ((ILESSP XFACTOR 101)
    (SETQ DELTAX 100)
    (SETQ XROUND (IQUOTIENT XFACTOR 2)))
  (T (SETQ DELTAX XFACTOR)
    (SETQ XROUND 50)))
(COND
  ((ILESSP YFACTOR 101)
    (SETQ DELTAY 100)
    (SETQ YROUND (IQUOTIENT YFACTOR 2)))
  (T (SETQ DELTAY YFACTOR)
    (SETQ YROUND 50)))
(SETQ NEWBITMAP (BITMAPCREATE (IQUOTIENT (IPLUS XROUND DELTAX (ITIMES (SUB1 BITMAPWIDTH)
                                          XFACTOR))
                                     100)
  (IQUOTIENT (IPLUS YROUND DELTAY (ITIMES (SUB1 BITMAPHEIGHT)
                                          YFACTOR))
            100)
  1))
(* MAKE ALL VALUES QUICKLY AVAILABLE)
(SETQ HEIGHT-1 (SUB1 BITMAPHEIGHT))
(SETQ RASTERWIDTH (fetch (BITMAP BITMAPRASTERWIDTH) of BITMAP))
(SETQ BITMAPBASE (fetch (BITMAP BITMAPBASE) of BITMAP)) (* AND THE NEW BITMAP VALUES)
(SETQ NEWHEIGHT-1 (SUB1 (fetch (BITMAP BITMAPHEIGHT) of NEWBITMAP)))
(SETQ NEWRASTERWIDTH (fetch (BITMAP BITMAPRASTERWIDTH) of NEWBITMAP))
(SETQ NEWBITMAPBASE (fetch (BITMAP BITMAPBASE) of NEWBITMAP))
(* OK, CRANK IT OUT)
(* ORIGWORD AND NEWWORD ARE SORTA CACHED FOR
SPEED PURPOSES)
[for Y from 0 to HEIGHT-1
  do [SETQ ORIGBASE (\ADDBASE BITMAPBASE (ITIMES RASTERWIDTH (IDIFFERENCE HEIGHT-1 Y)
    (SETQ ONLINE NIL)
    [for X from 0 to (SUB1 BITMAPWIDTH)
      do [AND (ZEROP (IMOD X 16))
        (SETQ ORIGWORD (\GETBASE ORIGBASE (LRSH X 4)
          (* LOOK FOR STRINGS OF "ON" BITS;
          THEN TREAT AS A LINE FOR TRANSLATIONAL PURPOSES)
(COND
  [(BITTEST ORIGWORD (\WORDELT BITMASKARRAY (IMOD X 16)))
    (OR ONLINE (AND (SETQ ONLINE T)
      (SETQ XSTART X)
      (SETQ YSTART Y)
      (* JUST SKIP OVER BLANKS)
    )
  (T
    (* SPELL THIS ALL OUT SO I CAN SEE WHAT'S GOIN' ON
    HERE)
    (SETQ XSTART (IQUOTIENT (IPLUS XROUND (ITIMES XSTART XFACTOR))
      100))
    (SETQ ENDY (IQUOTIENT (IPLUS (ITIMES YSTART YFACTOR)
      YROUND DELTAY)
      100))
    (SETQ YSTART (IQUOTIENT (IPLUS YROUND (ITIMES YSTART YFACTOR))
      100))
    (SETQ ENDX (IQUOTIENT (IPLUS XROUND (ITIMES (SUB1 X)
      XFACTOR))
      100))
    (for NY from YSTART to (SUB1 ENDY)
      do (SETQ NEWWORD (\GETBASE [SETQ NEWBASE (\ADDBASE NEWBITMAPBASE
        (ITIMES NEWRASTERWIDTH
        (IDIFFERENCE NEWHEIGHT-1
        NY]
        (LRSH XSTART 4)))
      (for NX from XSTART to ENDX do [AND (ZEROP (IMOD NX 16))
        (SETQ NEWWORD (\GETBASE NEWBASE
          (LRSH NX 4)
          [SETQ NEWWORD (LOGOR NEWWORD
            (\WORDELT BITMASKARRAY
            (IMOD NX 16)
            (AND (ZEROP (IMOD (ADD1 NX)
              16))
              (\PUTBASE NEWBASE (LRSH NX 4)
                NEWWORD)))
            (\PUTBASE NEWBASE (LRSH ENDX 4)

```

```

                                NEWWORD))
                                (SETQ ONLINE NIL]
(COND
  (ONLINE
    OF BITMAP)
    (* GOTTA CLEANUP AFTER THE LAST CASE)
    (* THIS IN CASE WORKING ON A LINE THAT GOES TO END
    (* GAWD! WHAT A WASTE O SPACE THIS IS.
    FIX LATER)
    (SETQ XSTART (IQUOTIENT (IPLUS XROUND (ITIMES XSTART XFACTOR))
                            100))
    (SETQ ENDY (IQUOTIENT (IPLUS (ITIMES YSTART YFACTOR)
                                YROUND DELTAY)
                            100))
    (SETQ YSTART (IQUOTIENT (IPLUS YROUND (ITIMES YSTART YFACTOR))
                            100))
    (SETQ ENDX (IQUOTIENT (IPLUS XROUND (ITIMES (SUB1 BITMAPWIDTH)
                                                XFACTOR))
                            100))
    (for NY from YSTART to (SUB1 ENDY)
      do (SETQ NEWWORD (\GETBASE [SETQ NEWBASE (\ADDBASE NEWBITMAPBASE
                                                         (ITIMES NEWRASTERWIDTH
                                                         (IDIFFERENCE NEWHEIGHT-1 NY]
                                                         (LRSH XSTART 4)))
        (for NX from XSTART to ENDX do [AND (ZEROP (IMOD NX 16))
          (SETQ NEWWORD (\GETBASE NEWBASE
                                   (LRSH NX 4]
          [SETQ NEWWORD (LOGOR NEWWORD
                              (\WORDELT BITMASKARRAY
                              (IMOD NX 16]
          (AND (ZEROP (IMOD (ADD1 NX)
                            16))
            (\PUTBASE NEWBASE (LRSH NX 4)
                           NEWWORD)))
          (\PUTBASE NEWBASE (LRSH ENDX 4)
                           NEWWORD]
    (RETURN NEWBITMAP])

```

**(ACE.COMPILE.FRAME.ACE**

[LAMBDA (PREC SUCC VERTICAL THRESHOLD)

(\* MJD "28-May-86 17:14")

(\* Calls the frame compiler on PREC and SUCC (bitmaps); returns the changes required to go from PREC to SUCC;  
supplies defaults if not given. VERTICAL and THRESHOLD are special args to the compiler)

(\* NEXT TWO ARE ARBITRARY DEFAULTS)

```

(OR VERTICAL (SETQ VERTICAL 16))
(OR THRESHOLD (SETQ THRESHOLD 50))
(ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
  "COMPILING")

```

(\* ACE.COMPILE.FRAME PREC SUCC VERTICAL THRESHOLD)

```

(SETQ ACE.COMPIER (ADD.PROCESS (BQUOTE (ACE.COMPILE.FRAME , PREC , SUCC , VERTICAL , THRESHOLD]))

```

**(ACE.MM1201.INIT**

[LAMBDA (INIT?)

(\* PmT "24-Apr-85 17:16")

(\* Inits the RS232 port and the MM1201 graphics tablet; If INIT? = ASK then the user is asked if s/he wants to init;  
If = T then else auto init; else just the tablet is initialized)

(\* DEFAULT RS232 IS 4800; DOESN'T QUITE WORK AT 9600)

```

(ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
  "INIT TABLET")

```

[COND

```

  ((OR (EQ INIT? T)
    (AND (EQ INIT? (QUOTE ASK))
      (ACE.CONFIRMIT "Click LEFT to Initialize Tablet." T)))
    (RS232INIT (COND

```

```

      ((KEYDOWNP (QUOTE LSHIFT))

```

```

        (OR (MENU (create MENU

```

```

          ITEMS _ (QUOTE (9600 4800 2400 1200 600 300 150 75))

```

```

          TITLE _ "Select Baud Rate"

```

```

          CENTERFLG _ T))

```

```

        4800))

```

```

      (T 4800))

```

```

      8 NIL 1)

```

```

    (forDuration 50 timerUnits (QUOTE MILLISECONDS))

```

```

    (RS232CLEARBUFFER (QUOTE BOTH))

```

```

    (RS232WRITEBYTE 32 T)

```

```

    (forDuration 50 timerUnits (QUOTE MILLISECONDS))

```

```

  (RS232CLEARBUFFER (QUOTE BOTH))

```

```

  (RS232WRITEBYTE 114)

```

```

  (RS232WRITEBYTE (LOGAND SCREENWIDTH 255))

```

```

  (RS232WRITEBYTE (LRSH SCREENWIDTH 8))

```

```

  (RS232WRITEBYTE (LOGAND SCREENHEIGHT 255))

```

```

  (RS232WRITEBYTE (LRSH SCREENHEIGHT 8)

```

```

    T)

```

```

  (forDuration 50 timerUnits (QUOTE MILLISECONDS))

```

```

  (RS232WRITEBYTE 68 T)

```

```

])

```

(\* set tablet baud rate:)

(\* Set x and y scale factors: USE SCREEN SIZE  
(Scaling to window size is an interesting idea))

(\* SET TABLET FOR POLLING MODE)





[illegible]

```
{MEDLEY}<lispusers>ACE>ACE-EDIT.;1
```

Page 19

```
(RPAQQ RS232LOSTCHARFN RS232LOSTCHARFN)
```

```
( * * MENUS FOR ACE-EDIT)
```

```
(DECLARE: DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS ACE.CONTROL.WINDOW ACE.EDIT.FRAME.MENU ACE.EDIT.LINEART.ADJ.MENU ACE.EDIT.MOVE.MENU  
  ACE.EDIT.TEXT.FONT.MENU ACE.EDIT.TEXT.SIZE.MENU ACE.EDIT.TEXT.FACE.MENU ACE.EDIT.TEXTURE.MENU  
  ACE.EDIT.PUTDOWN.MENU)  
)
```

```
(ACE.EDIT.CLEAR.ALL.MENUS)
```

```
(PUTPROPS ACE-EDIT COPYRIGHT ("Xerox Corporation" 1986))
```

---

### FUNCTION INDEX

ACE.COMPILE.FRAME.ACE .....	16	ACE.EDIT.PAINT .....	8	ACE.MM1201.INIT .....	16
ACE.EDIT .....	1	ACE.EDIT.POINT&CODE .....	13	ACE.MM1201.PROBLEM .....	17
ACE.EDIT.CLEAR.ALL.MENUS .....	17	ACE.EDIT.PUTDOWN.BITMAP .....	12	ACE.MM1201.POLL .....	17
ACE.EDIT.COMBINE.REGION .....	6	ACE.EDIT.REDRAW.ABITMAP .....	14	ACE.READBRUSHSHAPE .....	13
ACE.EDIT.CREATE.MENU.TEXTURES ...	11	ACE.EDIT.SETUP.EDIT.MENU .....	2	ACE.SCALE.BITMAP .....	14
ACE.EDIT.FRAME .....	2	ACE.EDIT.TEXT .....	6	ACE.SCANLINESEEDFILL .....	11
ACE.EDIT.LINEART .....	3	ACE.EDIT.TEXTURE.AREA .....	8	ACEEDITBM .....	12
ACE.EDIT.LINEART.ADJ .....	4	ACE.EDIT.TEXTURE.REGION .....	7	ACEGETREGIONFACTOR .....	3
ACE.EDIT.LINEART.DRAW .....	4	ACE.EXTRACTBM .....	14	ACEROTATEREGION .....	3
ACE.EDIT.LINEART.TRACKLINE .....	5	ACE.FILLWITHTEXTURE .....	10	ACESCALEREGION .....	3
ACE.EDIT.MOVE.REGION .....	5	ACE.GET.DEVICE.STATE .....	13	ROTATEBM .....	17
ACE.EDIT.MOVE.REGION.ASST .....	12	ACE.GET.DEVICE.STATE&CURSOR .....	14	RS232LOSTCHARFN .....	18

---

### VARIABLE INDEX

ACE.EDIT.LINEART.SQUARE.CURSOR ..	18	ACELOGOMAP .....	18	RS232LOSTCHARFN .....	19
-----------------------------------	----	------------------	----	-----------------------	----

---

### MACRO INDEX

ACE.POPPOS .....	18	ACE.PUSHPOS .....	18
------------------	----	-------------------	----

---