```
(RPAQQ ACE-MAINCOMS
       [(* MAIN TOP LEVEL STUFF)
        (FNS ACE ACE.ANIMATE ACE.RUN ACEGETFRAME# ACERUNLOOP ACE.NEW.SEQUENCE ACE.NEW.FRAME ACE.QUIT.ACE
             ACE.RESET.SEQ ACE.RUN.CURRENT.SEQ ACE.DELAY ACE.DELAY.FRAME ACE.DELAY.SEQ ACE.DECREMENT.FRAME
             ACE.INCREMENT.FRAME ACE.DELETE.FRAME ACE.SET.DEVICE ACE.QUICKDRAW&UPD ACE.RECONSTRUCT.FRAME SUBLIST
             )
        (* TRILLIUM STUFF)
        (FNS ACE.TRILLIUM ACE.TRILLIUM.LOOP ACE.RUN.TRILLIUM ACE.QUIT.TRILLIUM ACE.CREATE.EDITING.BORDER)
        (* I/O STUFF)
        (FNS ACE.GET.SEQ.FILE ACE.PUT.SEQ.FILE ACE.GET.A.FILE.NAME)
        (* HELPER FNS)
        (FNS ACE.ASKEM ACE.TELLEM ACE.CONFIRMIT ACE.DEFINE.SEQ.WINDOW ACE.FIGURE.OUT.WINDOW
             ACE.RETURN.CLOSEST.VERTEX ACE.NEW.SEQ.ASST ACE.DELAY.FRAME.ASST ACE.SETUP.CW.CLIPPING.REGIONS
             ACE.CHECKSTUFF ACE.UPD.CONTROL.WINDOW ACE.UPD.CW.MULE ACE.UPD.CLEAR.SET.LINE
             ACE.CREATE.CONTROL.MENU ACE.SEQ.FETCH.WIDTH ACE.SEQ.FETCH.HEIGHT ACE.SET.SEQ.CLIP.REGION ACE.ASKEM2
             ACE.TELLEM2 ACE.UPD.CONTROL.WINDOW2)
        (* The following Macros set up restricting clipping regions)
        (MACROS ACE.MAC.CW.INFO.CLIP ACE.MAC.CW.PROMPT.CLIP ACE.MAC.SEQ.CLIP)
        (MACROS ACE.MAC.FETCH.WIDTH ACE.MAC.FETCH.HEIGHT)
        (CURSORS ACE.LEFTMOUSE.CURSOR ACE.MIDDLEMOUSE.CURSOR ACE.RIGHTMOUSE.CURSOR ACE.ALLMOUSE.CURSOR)
        (GLOBALVARS ACE.CONTROL.WINDOW ACE.DIRECTORY ACE.SEQ.WINDOW ACE.SEQ.WIDTH ACE.SEQ.HEIGHT
             ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF ACE.CURRENT.SEQUENCE ACE.CURRENT.SEQUENCE.NAME
             ACE.FRAME.TAIL ACE.CURRENT.FRAME ACE.VERTICAL.BLOCK ACE.AREA.THRESHOLD ACE.RUNNING.UNDER.TRILLIUM
             ACE.LEFTMOUSE.CURSOR ACE.MIDDLEMOUSE.CURSOR ACE.RIGHTMOUSE.CURSOR ACE.ALLMOUSE.CURSOR)
        (* MENUS IN MAIN)
        (GLOBALVARS ACE.CONTROL.MENU ACE.DELAY.MENU ACE.SET.DEVICE.MENU)
        (P (SETQ ACE.CONTROL.WINDOW NIL)
           (SETQ ACE.CONTROL.MENU NIL)
           (SETQ ACE.DELAY.MENU NIL)
           (SETQ ACE.SET.DEVICE.MENU NIL))
        (DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                               (NLAML)
                                                                               (LAMA])


        (* * MAIN TOP LEVEL STUFF)


(DEFINEQ

(ACE
  [LAMBDA (SEQUENCE WINDOW POSITION APPLICATION)                    (* MD "14-Jun-85 17:34")

          (* Top level function to run animation. All ARGs are optional (No ARGs means just run "normal" ACE);
          Current APPLICATIONs are NIL (Normal) and TRILLIUM. If TRILLIUM, then POSITION is necessary
          (and WINDOW very highly reccommended!); ACE creates ACE.CONTROL.WINDOW and menu if necessary;
          then decides about APPLICATION)

    (PROG (FONT TEMP.REGION)
          (OR ACE.CONTROL.MENU (ACE.CREATE.CONTROL.MENU))
          (SETQ FONT (FONTCREATE (QUOTE HELVETICA)
                                 10
                                 (QUOTE BOLD)))
          (COND
             ((WINDOWP ACE.CONTROL.WINDOW)
              (TOTOPW ACE.CONTROL.WINDOW))
             (T (SETQ ACE.CONTROL.WINDOW (CREATEW (LIST 500 500 (IPLUS 2 (fetch IMAGEWIDTH of ACE.CONTROL.MENU))
                                                        (IPLUS (ITIMES 4 (FONTPROP FONT (QUOTE HEIGHT)))
                                                               (fetch IMAGEHEIGHT of ACE.CONTROL.MENU)
                                                               25))
                                                  (CONCAT "ACE  v. " "2.1" "      Control Window")
                                                  1))
                (DSPFONT FONT ACE.CONTROL.WINDOW)
                (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE RESHAPEFN)
                       (QUOTE DON'T))
                (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE REPAINTFN)
                       (QUOTE DON'T))
                (ACE.SETUP.CW.CLIPPING.REGIONS ACE.CONTROL.MENU)))
          (ACE.TELLEM "Position This Window" T)
          (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE MOVEFN)
                 NIL)
          (MOVEW ACE.CONTROL.WINDOW)
          (SETQ ACE.AREA.THRESHOLD 50)
          (SETQ ACE.VERTICAL.BLOCK 16)
          (SETQ ACE.CURRENT.SEQUENCE.NAME NIL)
```

```
      (COND
        ((EQ APPLICATION (QUOTE TRILLIUM))
         (RETURN (ACE.TRILLIUM WINDOW SEQUENCE POSITION)))
        (T (SETQ ACE.RUNNING.UNDER.TRILLIUM NIL)
           (SETQ ACE.DIRECTORY (ACE.ASKEM "Animation Directory? " T LOGINHOST/DIR))
           (ACE.TELLEM "If using the Tablet," T)
           (ACE.TELLEM "Be sure and Initialize it." (QUOTE L))
           (SETQ ACE.SEQ.WINDOW NIL)
           (SETQ ACE.SEQ.WIDTH NIL)
           (SETQ ACE.SEQ.HEIGHT NIL)
           (SETQ ACE.SEQ.WINDOW.XOFF 0)
           (SETQ ACE.SEQ.WINDOW.YOFF 0)
           (SETQ ACE.CURRENT.SEQUENCE NIL)
           (SETQ ACE.FRAME.TAIL NIL)
           (SETQ ACE.CURRENT.FRAME NIL)
           (replace (MENU WHENSELECTEDFN) of ACE.CONTROL.MENU with (QUOTE ACE.ANIMATE))
           (ADDMENU ACE.CONTROL.MENU ACE.CONTROL.WINDOW (CONS 0 0))
           (ACE.UPD.CONTROL.WINDOW (QUOTE RESET])
```

(**ACE.ANIMATE**
```
  [LAMBDA (ITEM WHO CARES)                                             (* MD "26-Jun-85 14:21")

          (* When running "normal" ACE, this is the WHENSELECTEDFN for deciding what to do.
          ITEM is the only ARG of significance (WHO & CARES just to keep MENU package happy);
          "TRILLIUM ACE" has its own loop thingy)

    (ACE.TELLEM "" T)
    (SELECTQ (CADADR ITEM)
        (NIL NIL)
        (FRAME (ACE.NEW.FRAME))
        (EDIT (ACE.EDIT))
        (RUN (ACE.RUN.CURRENT.SEQ))
        (LOOP (ACERUNLOOP 1 (QUOTE END)))
        (SUBLOOP (ACEGETFRAME#))
        (NEW (ACE.NEW.SEQUENCE))
        (I/O:GET (ACE.GET.SEQ.FILE))
        (I/O:PUT (ACE.PUT.SEQ.FILE))
        (DEL (ACE.DELETE.FRAME))
        (RESET (ACE.RESET.SEQ))
        (INC (ACE.INCREMENT.FRAME))
        (DEC (ACE.DECREMENT.FRAME))
        (TIME (ACE.DELAY))
        (INIT.MM1201 (ACE.MM1201.INIT T))
        (DEVICE (ACE.SET.DEVICE))
        (ACESETTHRESHOLD
            (ACESETTHRESHOLD))
        (QUIT (ACE.QUIT.ACE))
        NIL)
    (AND (OPENWP ACE.CONTROL.WINDOW)
        (ACE.UPD.CONTROL.WINDOW (QUOTE UPD)))
    (TTY.PROCESS T])
```

(**ACE.RUN**
```
  [LAMBDA (SEQ WINDOW XOFFSET YOFFSET TIMER)              (* PmT "18-Apr-85 18:21")

          (* This runs animation sequences. Simple, eh? Gots to have a SEQ and a WINDOW to show it in, and where in the window
          to show it (i.e. XOFFSET YOFFSET); TIMER is wholly unecessary NOTE: this thang doesn't use any clipping region, just an
          offset)

    (for FRAME in SEQ do ((SETQ TIMER (SETUPTIMER (fetch (ACE.FRAME DELAY) of FRAME)
                                            TIMER
                                            (QUOTE MILLISECONDS)))
                    (for FRAME.PART in (fetch (ACE.FRAME BLITS) of FRAME)
                        do (BITBLT (fetch (ACE.BLIT BITMAP) of FRAME.PART)
                                NIL NIL WINDOW (IPLUS XOFFSET (fetch (ACE.BLIT XCOOR) of FRAME.PART))
                                (IPLUS YOFFSET (fetch (ACE.BLIT YCOOR) of FRAME.PART))
                                NIL NIL (QUOTE INPUT)
                                (QUOTE REPLACE)))
                    (until (TIMEREXPIRED? TIMER (QUOTE MILLISECONDS)) do])
```

(**ACEGETFRAME#**
```
  [LAMBDA NIL                                                          (* MD "27-Jun-85 12:51")
    (ACERUNLOOP (RNUMBER "From frame #")
          (RNUMBER "To frame #:"])
```

(**ACERUNLOOP**
```
  [LAMBDA (START END)                                                 (* MD "26-Jun-85 17:07")
    (if (EQ END (QUOTE END))
        then (SETQ END (LENGTH ACE.CURRENT.SEQUENCE)))
    [ACE.QUICKDRAW&UPD (LIST (CAR (NTH ACE.CURRENT.SEQUENCE START]
    (until (KEYDOWNP (QUOTE SPACE)) do (ACE.RUN (SUBLIST ACE.CURRENT.SEQUENCE START END)
                                            ACE.SEQ.WINDOW ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF])
```

(ACE.NEW.SEQUENCE
```
  [LAMBDA (REGION)                                      (* PmT "30-Apr-85 16:21")
    (PROG (TEMP.REGION)
          (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
                 "NEW")
          (COND
              ((OR (NULL ACE.CURRENT.SEQUENCE)
                   (ACE.TELLEM "Creating a NEW Sequence will ERASE" T)
                   (ACE.TELLEM "the Current Animation Sequence." (QUOTE L))
                   (ACE.CONFIRMIT "Click LEFT to Create NEW Sequence." (QUOTE L)))
              [COND
                 ((REGIONP REGION)
                  (SETQ TEMP.REGION REGION))
                 (T (ACE.TELLEM "Specify SIZE of the NEW Sequence." T)
                    (ACE.TELLEM "Watch ACE window for COORs." (QUOTE L))
                    (SETQ TEMP.REGION (GETREGION NIL NIL NIL (QUOTE ACE.NEW.SEQ.ASST)
                                                 NIL]
              (SETQ ACE.SEQ.WIDTH (fetch (REGION WIDTH) of TEMP.REGION))
              (SETQ ACE.SEQ.HEIGHT (fetch (REGION HEIGHT) of TEMP.REGION))
              [SETQ ACE.CURRENT.SEQUENCE (LIST (create ACE.FRAME
                                                      DELAY _ 0
                                                      BLITS _ (LIST (create ACE.BLIT
                                                                            BITMAP _ (BITMAPCREATE ACE.SEQ.WIDTH
                                                                                          ACE.SEQ.HEIGHT 1)
                                                                            XCOOR _ 0
                                                                            YCOOR _ 0]
              (ACE.FIGURE.OUT.WINDOW)
              (ACE.QUICKDRAW&UPD (LIST (CAR ACE.CURRENT.SEQUENCE]
```

(ACE.NEW.FRAME
```
  [LAMBDA NIL                                           (* PmT "24-Jan-85 13:22")
    (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
           "FRAME")
    (COND
        ([NULL (AND (ACE.CHECKSTUFF (QUOTE SEQ))
                    (OR (ACE.CHECKSTUFF (QUOTE FRAME))
                        (ACE.TELLEM "Can't put a frame before the First Frame." T]
         NIL)
        (T (RPLACD ACE.CURRENT.FRAME (CONS (create ACE.FRAME
                                                  DELAY _ 0
                                                  BLITS _ NIL)
                                          (CDR ACE.CURRENT.FRAME)))
           (SETQ ACE.CURRENT.FRAME (CDR ACE.CURRENT.FRAME))
           (SETQ ACE.FRAME.TAIL (CDR ACE.CURRENT.FRAME))
           (ACE.TELLEM "Going to EDIT ..." T)
           (ACE.EDIT])
```

(ACE.QUIT.ACE
```
  [LAMBDA NIL                                           (* MD "24-Jun-85 14:49")
    (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
           "QUIT")
    (ACE.TELLEM "QUITing will Close All Animation Windows;" T)
    (ACE.TELLEM "All Images and Data will be LOST." (QUOTE L))
    (COND
        ((ACE.CONFIRMIT "Click LEFT to QUIT." (QUOTE L))
         (CLOSEW ACE.SEQ.WINDOW)
         (CLOSEW ACE.CONTROL.WINDOW)
         ACE.CURRENT.SEQUENCE)
        (T
            (* MAKE THIS BE SOME KINDA ICON IN THE FUTURE; ASK: DO YOU WANT TO QUIT COMPLETELY OR JUST STOP
            FOR A WHILE)

            (ACE.TELLEM "QUIT Aborted." T])
```

(ACE.RESET.SEQ
```
  [LAMBDA NIL                                           (* PmT "30-Apr-85 16:37")
    (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
           "RESET")
    (COND
        [(AND ACE.SEQ.WINDOW (ACE.CHECKSTUFF (QUOTE SEQ)))
         (ACE.QUICKDRAW&UPD (LIST (CAR ACE.CURRENT.SEQUENCE]
        (T (ACE.TELLEM "There is No Current Sequence." T])
```

(ACE.RUN.CURRENT.SEQ
```
  [LAMBDA NIL                                           (* PmT "18-Apr-85 18:23")
                                                        (* just a pretty interface to ACE.RUN)

    (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
           "RUN")
    (COND
        ((ACE.CHECKSTUFF (QUOTE SEQ))
         (RECLAIM)
         (ACE.RUN ACE.FRAME.TAIL ACE.SEQ.WINDOW ACE.SEQ.WINDOW.XOFF ACE.SEQ.WINDOW.YOFF)
```

```
            (SETQ ACE.FRAME.TAIL NIL)
            (SETQ ACE.CURRENT.FRAME (LAST ACE.CURRENT.SEQUENCE])
```

(**ACE.DELAY**
```
  [LAMBDA NIL                                                  (* PmT " 2-May-85 20:53")
                                                               (* For setting delays (in MSECs) between frames.
                                                               Lots of work needed here; esp. delay in-betweening)

      (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
             "DELAY")
      (SELECTQ [MENU (OR ACE.DELAY.MENU (SETQ ACE.DELAY.MENU (create MENU
                                                                 ITEMS _ (QUOTE (("Set Delay on a Frame"
                                                                          (QUOTE FDELAY)
                                                                          "Adjust the delay on any
                                                                          frame by number")
                                                                         ("Reset Entire Sequence"
                                                                          (QUOTE SDELAY)
                                                                          "Set the delay on every
                                                                          frame")))
                                                                 CENTERFLG _ T
                                                                 TITLE _ "Sequence Timing Adjustments"]
             (NIL NIL)
             (SDELAY (ACE.DELAY.SEQ))
             (FDELAY (ACE.DELAY.FRAME))
             NIL])
```

(**ACE.DELAY.FRAME**
```
  [LAMBDA NIL                                                  (* MD "21-Jun-85 14:14")
      (PROG (CHOICE FRAME)
        LOOP
             [SETQ CHOICE (MENU (create MENU
                                         ITEMS _ (NCONC1 (for FRAME in ACE.CURRENT.SEQUENCE
                                                            bind (COUNT _ 0)
                                                            collect ((SETQ COUNT (ADD1 COUNT))
                                                                       (LIST (CONCAT "Frame " COUNT " : "
                                                                                (fetch (ACE.FRAME DELAY) of FRAME))
                                                                             COUNT)))
                                                    (QUOTE (Quit (QUOTE QUIT)
                                                                 "Stop adjusting delays")))
                                         TITLE _ "Frame Delays"
                                         WHENHELDFN _ (QUOTE ACE.DELAY.FRAME.ASST]
             [COND
                ((NULL CHOICE)
                 NIL)
                ((EQ CHOICE (QUOTE QUIT))
                 (RETURN NIL))
                (T (SETQ FRAME (CAR (NTH ACE.CURRENT.SEQUENCE CHOICE)))
                   (AND FRAME (replace (ACE.FRAME DELAY) of FRAME with (SETQ CHOICE (RNUMBER (CONCAT "Frame " CHOICE
                                                                                        "; New Delay: "]
             (GO LOOP])
```

(**ACE.DELAY.SEQ**
```
  [LAMBDA NIL                                                  (* MD "21-Jun-85 14:31")
      (PROG (NEW.DELAY.VALUE)
            (COND
               [(FIXP (SETQ NEW.DELAY.VALUE (RNUMBER "Delay for entire  sequence:"]
               (T (RETURN NIL)))
            (for FRAME in ACE.CURRENT.SEQUENCE do (replace (ACE.FRAME DELAY) of FRAME with NEW.DELAY.VALUE])
```

(**ACE.DECREMENT.FRAME**
```
  [LAMBDA NIL                                                  (* PmT "21-Dec-84 14:12")
      (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
             "DEC")
      (AND (ACE.CHECKSTUFF (QUOTE SEQ))
           (ACE.CHECKSTUFF (QUOTE FRAME))
           (ACE.QUICKDRAW&UPD (LDIFF ACE.CURRENT.SEQUENCE ACE.CURRENT.FRAME])
```

(**ACE.INCREMENT.FRAME**
```
  [LAMBDA NIL                                                  (* MD "18-Jun-85 16:12")
      (PROG (CUR.FRAME)
            (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
                   "INC")
            (COND
               ([NULL (AND (ACE.CHECKSTUFF (QUOTE SEQ))
                           (ACE.CHECKSTUFF (QUOTE TAIL]
                 NIL)
               (T (SETQ CUR.FRAME (CAR ACE.FRAME.TAIL))
                  (SETQ ACE.CURRENT.FRAME ACE.FRAME.TAIL)
                  (SETQ ACE.FRAME.TAIL (CDR ACE.FRAME.TAIL))
                  (COND
                     ((NULL (fetch (ACE.FRAME BLITS) of CUR.FRAME))
                      NIL)
                     (T (ACE.MAC.SEQ.CLIP (for FRAME.PART in (fetch (ACE.FRAME BLITS) of CUR.FRAME)
```

```
                                                      do (BITBLT (fetch (ACE.BLIT BITMAP) of FRAME.PART)
                                                          NIL NIL ACE.SEQ.WINDOW (IPLUS ACE.SEQ.WINDOW.XOFF
                                                                                      (fetch (ACE.BLIT XCOOR)
                                                                                          of FRAME.PART))
                                                          (IPLUS ACE.SEQ.WINDOW.YOFF (fetch (ACE.BLIT YCOOR)
                                                                                          of FRAME.PART))
                                                      NIL NIL (QUOTE INPUT)
                                                      (QUOTE REPLACE])
```

(**ACE.DELETE.FRAME**
```
  [LAMBDA NIL                                                        (* PmT "24-Apr-85 14:19")
                                                                     (* Deletes the current frame; recompiles the previous frame with

      the successor frame)
    (PROG (BEFORE.BM AFTER.BM)
          (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
              "DELETE")
          (COND
             ([NULL (AND (ACE.CHECKSTUFF (QUOTE SEQ))
                          (ACE.CHECKSTUFF (QUOTE FRAME]
              NIL)
             ((EQ ACE.CURRENT.SEQUENCE ACE.CURRENT.FRAME)
              (ACE.TELLEM "Can't DELETE first frame.  Aborted." T))
             ((NULL (ACE.CONFIRMIT "Click LEFT to Confirm Delete" T))
              NIL)
             ((NULL ACE.FRAME.TAIL)
              (SETQ ACE.CURRENT.SEQUENCE (LDIFF ACE.CURRENT.SEQUENCE ACE.CURRENT.FRAME))
              (ACE.QUICKDRAW&UPD ACE.CURRENT.SEQUENCE))
             (T (SETQ BEFORE.BM (ACE.RECONSTRUCT.FRAME (LDIFF ACE.CURRENT.SEQUENCE ACE.CURRENT.FRAME)))
                [SETQ AFTER.BM (ACE.RECONSTRUCT.FRAME (LDIFF ACE.CURRENT.SEQUENCE (CDR ACE.FRAME.TAIL]
                (replace (ACE.FRAME BLITS) of (CAR ACE.CURRENT.FRAME) with (ACE.COMPILE.FRAME BEFORE.BM AFTER.BM
                                                                            ACE.VERTICAL.BLOCK
                                                                            ACE.AREA.THRESHOLD))
                (RPLACD ACE.CURRENT.FRAME (CDR ACE.FRAME.TAIL))
                (ACE.QUICKDRAW&UPD (LDIFF ACE.CURRENT.SEQUENCE (CDR ACE.CURRENT.FRAME])
```

(**ACE.SET.DEVICE**
```
  [LAMBDA NIL                                                        (* PmT "23-Apr-85 13:44")
                                                                     (* Selects MOUSE or TABLET as the primary input device)

      (ACE.UPD.CONTROL.WINDOW (QUOTE OPERATION)
          "SET DEVICE")
      (ACE.UPD.CONTROL.WINDOW (QUOTE DEVICE))
      (ACE.UPD.CONTROL.WINDOW (QUOTE DEVICE)
          (MENU (OR ACE.SET.DEVICE.MENU (SETQ ACE.SET.DEVICE.MENU (create MENU
                                                            ITEMS _ (QUOTE (("Mouse" (QUOTE MOUSE)
                                                                                    "Use the
                                                                                    standard mouse
                                                                                    for drawing and
                                                                                    such")
                                                                            ("Tablet" (QUOTE
                                                                                        MM1201
                                                                                        )
                                                                                    "Use the MM1201
                                                                                    Tablet as the
                                                                                    input device"))
                                                                    )
                                                            TITLE _ "Select Input Device"
                                                            CENTERFLG _ T])
```

(**ACE.QUICKDRAW&UPD**
```
  [LAMBDA (PARTIAL.SEQ)                                              (* PmT "30-Apr-85 16:11")

          (* Updates the frame showing in the A.S.Window and update sequence pointers and stuff.
           PARTIAL.SEQ is a list of frames to show; The last frame in PARTIAL.SEQ becomes the new current frame)

      (COND
         (PARTIAL.SEQ [ACE.MAC.SEQ.CLIP (for FRAME in PARTIAL.SEQ
                                          do (COND
                                                ((NULL (fetch (ACE.FRAME BLITS) of FRAME))
                                                 NIL)
                                                (T (for FRAME.PART in (fetch (ACE.FRAME BLITS) of FRAME)
                                                        do (BITBLT (fetch (ACE.BLIT BITMAP) of FRAME.PART)
                                                            NIL NIL ACE.SEQ.WINDOW (IPLUS ACE.SEQ.WINDOW.XOFF
                                                                                        (fetch (ACE.BLIT
                                                                                                XCOOR)
                                                                                            of FRAME.PART))
                                                            (IPLUS ACE.SEQ.WINDOW.YOFF (fetch (ACE.BLIT YCOOR)
                                                                                            of FRAME.PART))
                                                            NIL NIL (QUOTE INPUT)
                                                            (QUOTE REPLACE]
                        (SETQ ACE.CURRENT.FRAME ACE.CURRENT.SEQUENCE)
                        (for X from 1 to (SUB1 (LENGTH PARTIAL.SEQ)) do (SETQ ACE.CURRENT.FRAME (CDR ACE.CURRENT.FRAME)))
                        (SETQ ACE.FRAME.TAIL (CDR ACE.CURRENT.FRAME])
```

(ACE.RECONSTRUCT.FRAME
```
  [LAMBDA (SEQ)                                                    (* PmT "18-Apr-85 18:54")

          (* Creates a bitmap out of SEQ; Essentially, the last virtual frame in SEQ is converted to a "real" frame and returned)

    (PROG (ABITMAP)
          [SETQ ABITMAP (BITMAPCOPY (fetch (ACE.BLIT BITMAP) of (CAR (fetch (ACE.FRAME BLITS) of (CAR SEQ]
          [for FRAME in (CDR SEQ) do (COND
                                       ((NULL (fetch (ACE.FRAME BLITS) of FRAME)))
                                       (T (for FRAME.PART in (fetch (ACE.FRAME BLITS) of FRAME)
                                             do (BITBLT (fetch (ACE.BLIT BITMAP) of FRAME.PART)
                                                        NIL NIL ABITMAP (fetch (ACE.BLIT XCOOR) of FRAME.PART)
                                                         (fetch (ACE.BLIT YCOOR) of FRAME.PART)
                                                        NIL NIL (QUOTE INPUT)
                                                        (QUOTE REPLACE]
          (RETURN ABITMAP])
```

(SUBLIST
```
  [LAMBDA (L M N)                                                  (* MD "26-Jun-85 16:00")
    (LDIFF (NTH L M)
           (NTH L (ADD1 N))))

)


          (* * TRILLIUM STUFF)

(DEFINEQ
```

(ACE.TRILLIUM
```
  [LAMBDA (WINDOW SEQUENCE POSITION)                               (* PmT "30-Apr-85 16:44")

          (* This here sets up stuff for running animation (functionally) from Trillium.
          ARGS: WINDOW is opt (but really should be given), SEQUENCE is opt, POSITION *MUST* be given;
          if not, ACE bags it (Trillium must always supply a place to put the animation!): This FN just checks args and sets the state of
          ACE, then calls the actual "polling" FN NOTE: This should only be called from ACE;
          take a look at ACE)

    (PROG NIL
          (SETQ ACE.RUNNING.UNDER.TRILLIUM T)
          (DISPLAY.FRAME CURRENT.FRAME)                            (* See if got a valid window or not)
          [COND
             ((WINDOWP WINDOW)
              (SETQ ACE.SEQ.WINDOW WINDOW))
             (T (ACE.TELLEM "There is no Window Specification" T)
                (COND
                   ((ACE.CONFIRMIT "Click LEFT to use Current Interface Window." (QUOTE L))
                    (SETQ ACE.SEQ.WINDOW CURRENT.INTERFACE.WINDOW))
                   (T (ACE.TELLEM "Error in Window Specification." T)
                      (ACE.TELLEM "Likely a Trillium error.  ACE aborted." (QUOTE L))
                      (ACE.CONFIRMIT "Click any button to Exit." (QUOTE L)
                             (QUOTE ANY))
                      (RETURN NIL]                                 (* The following *might* be used instead of a window error;
                                                                   you decide)
                                                                   (* SETQ ACE.SEQ.WINDOW (EVAL
                                                                   (ACE.ASKEM "Enter the Name of the Window: " T NIL 60)))
                                                                   (* Was given a valid postion?)
          (COND
             ((POSITIONP POSITION)
              (SETQ ACE.SEQ.WINDOW.XOFF (fetch (POSITION XCOORD) of POSITION))
              (SETQ ACE.SEQ.WINDOW.YOFF (fetch (POSITION YCOORD) of POSITION)))
             (T (ACE.TELLEM "No Position Specification.  Aborted." T)
                (ACE.TELLEM "This is likely a Trillium error." (QUOTE L))
                (ACE.CONFIRMIT "Click any button to Exit." (QUOTE L)
                       (QUOTE ANY))
                (RETURN NIL)))                                     (* Now check if given a sequence)
          (COND
             ((LISTP SEQUENCE)
              (SETQ ACE.CURRENT.SEQUENCE SEQUENCE)
              (SETQ ACE.CURRENT.FRAME ACE.CURRENT.SEQUENCE)
              (SETQ ACE.FRAME.TAIL (CDR ACE.CURRENT.FRAME))
              (SETQ ACE.SEQ.WIDTH (ACE.SEQ.FETCH.WIDTH))
              (SETQ ACE.SEQ.HEIGHT (ACE.SEQ.FETCH.HEIGHT))
              (ACE.CREATE.EDITING.BORDER)
              (ACE.SET.SEQ.CLIP.REGION))
             (T (SETQ ACE.CURRENT.SEQUENCE NIL)
                (SETQ ACE.FRAME.TAIL NIL)
                (SETQ ACE.CURRENT.FRAME NIL)))                     (* The WHENSELECTEDFN is different for Trillium;
                                                                   make it act like a regular old menu)
          (replace (MENU WHENSELECTEDFN) of ACE.CONTROL.MENU with (QUOTE DEFAULTWHENSELECTEDFN))
          (SETQ ACE.DIRECTORY (DIRECTORYNAME T T))
          (AND ACE.CURRENT.SEQUENCE (ACE.RESET.SEQ))
          (ACE.UPD.CONTROL.WINDOW (QUOTE RESET))                   (* Lock down window so menu coors only figured once;
                                                                   see ACE.TRILLIUM.LOOP)
          (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE MOVEFN)
                  (QUOTE DON'T))
```

```
        (RETURN (ACE.TRILLIUM.LOOP])
```

(**ACE.TRILLIUM.LOOP**
```
  [LAMBDA NIL                                                    (* PmT "18-Apr-85 18:41")
                                                                 (* This is the repeating loop for Trillium-Ace;
                                                                 just sits in here till QUIT)

    (PROG (CHOICE MENU.POS)
          [SETQ MENU.POS (CONS (DSPXOFFSET NIL (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE DSP)))
                               (DSPYOFFSET NIL (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE DSP]
      LOOP
          (ACE.TELLEM "" T)
          (SELECTQ (SETQ CHOICE (MENU ACE.CONTROL.MENU MENU.POS))
              (NIL NIL)
              (FRAME (ACE.NEW.FRAME))
              (EDIT (ACE.EDIT))
              (RUN (ACE.RUN.CURRENT.SEQ))
              (NEW (ACE.NEW.SEQUENCE))
              (I/O:GET (ACE.GET.SEQ.FILE))
              (I/O:PUT (ACE.PUT.SEQ.FILE))
              (DEL (ACE.DELETE.FRAME))
              (RESET (ACE.RESET.SEQ))
              (INC (ACE.INCREMENT.FRAME))
              (DEC (ACE.DECREMENT.FRAME))
              (TIME (ACE.DELAY))
              (INIT.MM1201 (ACE.MM1201.INIT T))
              (DEVICE (ACE.SET.DEVICE))
              (QUIT NIL)
              NIL)
          (ACE.UPD.CONTROL.WINDOW (QUOTE UPD))
          (OR (EQ CHOICE (QUOTE QUIT))
              (GO LOOP))
          (RETURN (ACE.QUIT.TRILLIUM])
```

(**ACE.RUN.TRILLIUM**
```
  [LAMBDA (SEQ WINDOW XOFFSET YOFFSET UPTO TIMER)                 (* PmT "18-Apr-85 18:45")

            (* Just like ACE.RUN except UPTO can be a FIXP denoting a frame;
            If UPTO is given, that frame is displayed (without delays); Good for initializing in Trillium)

    (COND
      [(NULL UPTO)
       (for FRAME in SEQ do ((SETQ TIMER (SETUPTIMER (fetch (ACE.FRAME DELAY) of FRAME)
                                            TIMER
                                            (QUOTE MILLISECONDS)))
                             (for FRAME.PART in (fetch (ACE.FRAME BLITS) of FRAME)
                                do (BITBLT (fetch (ACE.BLIT BITMAP) of FRAME.PART)
                                           NIL NIL WINDOW (IPLUS XOFFSET (fetch (ACE.BLIT XCOOR) of FRAME.PART))
                                           (IPLUS YOFFSET (fetch (ACE.BLIT YCOOR) of FRAME.PART))
                                           NIL NIL (QUOTE INPUT)
                                           (QUOTE REPLACE)))
                             (until (TIMEREXPIRED? TIMER (QUOTE MILLISECONDS)) do]
      ((AND (FIXP UPTO)
            (IGREATERP (ADD1 (LENGTH SEQ))
                   UPTO)
            (IGREATERP UPTO 0))
       (for FRAME in (LDIFF SEQ (NTH SEQ (ADD1 UPTO)))
          do (for FRAME.PART in (fetch (ACE.FRAME BLITS) of FRAME) do (BITBLT (fetch (ACE.BLIT BITMAP) of
                                                                                      FRAME.PART
                                                                                      )
                                                                       NIL NIL WINDOW
                                                                       (IPLUS XOFFSET (fetch (ACE.BLIT XCOOR)
                                                                                         of FRAME.PART))
                                                                       (IPLUS YOFFSET (fetch (ACE.BLIT YCOOR)
                                                                                         of FRAME.PART))
                                                                       NIL NIL (QUOTE INPUT)
                                                                       (QUOTE REPLACE])
```

(**ACE.QUIT.TRILLIUM**
```
  [LAMBDA NIL                                                     (* PmT "15-Mar-85 13:48")
    (CLOSEW ACE.CONTROL.WINDOW)
    (SETQ ACE.RUNNING.UNDER.TRILLIUM NIL)
    ACE.CURRENT.SEQUENCE])
```

(**ACE.CREATE.EDITING.BORDER**
```
  [LAMBDA (MODE)                                                  (* PmT "30-Apr-85 16:42")
    (PROG (X1 X2 Y1 Y2)
          (OR MODE (SETQ MODE (QUOTE PAINT)))
          (COND
             ((AND (NUMBERP ACE.SEQ.WIDTH)
                   (NUMBERP ACE.SEQ.HEIGHT))
              (SETQ X1 (IDIFFERENCE ACE.SEQ.WINDOW.XOFF 2))
              (SETQ X2 (IPLUS ACE.SEQ.WINDOW.XOFF ACE.SEQ.WIDTH))
              (SETQ Y1 (IDIFFERENCE ACE.SEQ.WINDOW.YOFF 2))
```

```
                (SETQ Y2 (IPLUS ACE.SEQ.WINDOW.YOFF ACE.SEQ.HEIGHT))
                (DRAWLINE X1 Y1 X1 Y2 2 MODE ACE.SEQ.WINDOW)
                (DRAWLINE X1 Y2 X2 Y2 2 MODE ACE.SEQ.WINDOW)
                (DRAWLINE X2 Y2 X2 Y1 2 MODE ACE.SEQ.WINDOW)
                (DRAWLINE X2 Y1 X1 Y1 2 MODE ACE.SEQ.WINDOW]

)


            (* * I/O STUFF)

(DEFINEQ

(**ACE.GET.SEQ.FILE**
  [LAMBDA NIL                                                     (* PmT "25-Apr-85 21:18")
                                                                  (* Gets an animation sequence. Resets
                                                                  ACE.CURRENT.SEQUENCE and the sequence clipping region)

    (RESETFORM (TTYDISPLAYSTREAM \TopLevelTtyWindow)
            (PROG (FILENAME TEMP.SEQUENCE.NAME)
                  (**ACE.UPD.CONTROL.WINDOW** (QUOTE OPERATION)
                        "GET FILE")
                  (OR (NULL ACE.CURRENT.SEQUENCE)
                      (**ACE.TELLEM** "Loading a Sequence will ERASE the Current" T)
                      (**ACE.CONFIRMIT** "Sequence; Click LEFT to confirm LOAD." (QUOTE L))
                      (**ACE.TELLEM** "Get Sequence ABORTED." T)
                      (RETURN NIL))
                  (SETQ FILENAME (**ACE.GET.A.FILE.NAME**))
                  [COND
                      ((NULL FILENAME)
                       (**ACE.TELLEM** "No NAME.  Aborted" T)
                       (RETURN NIL))
                      (T (SETQ FILENAME (PACKFILENAME (QUOTE BODY)
                                                FILENAME
                                                (QUOTE HOST)
                                                (FILENAMEFIELD ACE.DIRECTORY (QUOTE HOST))
                                                (QUOTE DIRECTORY)
                                                (FILENAMEFIELD ACE.DIRECTORY (QUOTE DIRECTORY]
                  (**ACE.TELLEM** "Loading: " T)
                  (**ACE.TELLEM** (CONCAT FILENAME " ... ")
                        (QUOTE L))
                  (RESETLST
                      [RESETSAVE (PROGN (CURSOR WAITINGCURSOR)
                                        (SETTOPVAL (QUOTE HELPFLAG)
                                                NIL))
                              (LIST (QUOTE PROGN)
                                    (LIST (QUOTE CURSOR)
                                          (QUOTE DEFAULTCURSOR))
                                    (LIST (QUOTE SETTOPVAL)
                                          (QUOTE (QUOTE HELPFLAG))
                                          (KWOTE (GETTOPVAL (QUOTE HELPFLAG]
                      [SETQ TEMP.SEQUENCE.NAME (CAR (ERRORSET (QUOTE (LOAD FILENAME (QUOTE SYSLOAD)))
                                                        (QUOTE NOBREAK])
                      (COND
                          (TEMP.SEQUENCE.NAME (SETQ ACE.CURRENT.SEQUENCE.NAME TEMP.SEQUENCE.NAME)
                                (SETQ ACE.SEQ.WIDTH (**ACE.SEQ.FETCH.WIDTH**))
                                (SETQ ACE.SEQ.HEIGHT (**ACE.SEQ.FETCH.HEIGHT**))
                                (**ACE.FIGURE.OUT.WINDOW**)
                                (**ACE.RESET.SEQ**))
                          (T (**ACE.TELLEM** "Not Found.")
                             (**ACE.TELLEM** "No Such File or File Server Problems." (QUOTE L])


(**ACE.PUT.SEQ.FILE**
  [LAMBDA NIL                                                     (* PmT " 2-May-85 20:50")
                                                                  (* Writes a sequence to a file; the file is NOT pretty printed)

    (PROG (FILENAME TEMP.SEQUENCE.NAME)
          (**ACE.UPD.CONTROL.WINDOW** (QUOTE OPERATION)
                "PUT FILE")
          [COND
              ((NULL (**ACE.CHECKSTUFF** (QUOTE SEQ)))
               (RETURN NIL))
              ((AND ACE.CURRENT.SEQUENCE.NAME (**ACE.CONFIRMIT** "Click LEFT to Keep Same Name." T))
               (SETQ FILENAME ACE.CURRENT.SEQUENCE.NAME))
              (T (SETQ FILENAME (**ACE.GET.A.FILE.NAME**))
                 (COND
                     ((NULL FILENAME)
                      (**ACE.TELLEM** "NIL ain't no good.  Aborted." T)
                      (RETURN NIL)))
                 (SETQ FILENAME (PACKFILENAME (QUOTE BODY)
                                        FILENAME
                                        (QUOTE HOST)
                                        (FILENAMEFIELD ACE.DIRECTORY (QUOTE HOST))
                                        (QUOTE DIRECTORY)
                                        (FILENAMEFIELD ACE.DIRECTORY (QUOTE DIRECTORY]
          (COND
              ((AND (FILENAMEFIELD FILENAME (QUOTE VERSION))
                    (NULL (**ACE.TELLEM** "Click LEFT to Write a New Version." T)))
```

```
                    (ACE.CONFIRMIT "Click any Other to Write Over Existing Version." (QUOTE L)))
                (SETQ FILENAME (PACKFILENAME (QUOTE VERSION)
                                        NIL
                                        (QUOTE BODY)
                                        FILENAME)))
            (T NIL))
        [SET (PACK* (FILENAMEFIELD FILENAME (QUOTE NAME))
                    (QUOTE COMS))
            (QUOTE ((UGLYVARS ACE.CURRENT.SEQUENCE]
        (PUTPROP (FILENAMEFIELD FILENAME (QUOTE NAME))
            (QUOTE FILETYPE)
            (QUOTE (DON'TLIST DON'TCOMPILE)))
        (ACE.TELLEM "Putting to File: " T)
        (ACE.TELLEM (CONCAT FILENAME " ... ")
            (QUOTE L))
        (RESETLST
            [RESETSAVE (PROGN (CURSOR WAITINGCURSOR)
                            (SETTOPVAL (QUOTE HELPFLAG)
                                    NIL))
                    (LIST (QUOTE PROGN)
                        (LIST (QUOTE CURSOR)
                            (QUOTE DEFAULTCURSOR))
                        (LIST (QUOTE SETTOPVAL)
                            (QUOTE (QUOTE HELPFLAG))
                            (KWOTE (GETTOPVAL (QUOTE HELPFLAG]
            [SETQ TEMP.SEQUENCE.NAME (CAR (ERRORSET [QUOTE (MAKEFILE FILENAME (QUOTE (NEW FAST]
                                                (QUOTE NOBREAK])
        (COND
            (TEMP.SEQUENCE.NAME (SETQ ACE.CURRENT.SEQUENCE.NAME TEMP.SEQUENCE.NAME)
                    (ACE.TELLEM "Done")
                    (DREMOVE (FILENAMEFIELD FILENAME (QUOTE NAME))
                        FILELST))
            (T (ACE.TELLEM "Aborted.")
                (ACE.TELLEM "Nothing doing.  Can't write this File out." (QUOTE L))
                (ACE.TELLEM "Check the Name.  Is the File Server Down?" (QUOTE L])
```

(**ACE.GET.A.FILE.NAME**
```
  [LAMBDA NIL                                                    (* PmT "18-Apr-85 19:44")
    (ACE.ASKEM "Enter FILENAME: " T NIL 120])
```

)

(* * HELPER FNS)

(DEFINEQ

(**ACE.ASKEM**
```
  [LAMBDA (STRING FLG DEFAULTANSWER TIMELIMIT? SPACES?)           (* MD "14-Jun-85 16:48")

            (* a prompting fn. STRING is the prompt string; FLG either T,L or NIL
            (T means clear before prompting, L means new line); DEFAULTANSWER just what it sounds like.
            TIMELIMIT? number or seconds to wait for answer (defaults to 120);
            if SPACES? is T, then the answer can have spaces in it)
                                                             (* TIMELIMIT? removed -
                                                             now waits forever. -
                                                             MJD)
    (ACE.MAC.CW.PROMPT.CLIP (PROGN (OR TIMELIMIT? (SETQ TIMELIMIT? 120))
                                (COND
                                    ((EQ FLG T)
                                    (DSPRESET ACE.CONTROL.WINDOW))
                                    ((EQ FLG (QUOTE L))
                                    (TERPRI ACE.CONTROL.WINDOW)))
                                (MKATOM (PROMPTFORWORD STRING DEFAULTANSWER NIL ACE.CONTROL.WINDOW NIL NIL
                                            (AND SPACES? (CHARCODE (EOL ESCAPE LF)))
```

(**ACE.TELLEM**
```
  [LAMBDA (STRING FLG)                                           (* PmT "23-Apr-85 13:49")

            (* Writes STRING in the A.C.W prompt region; FLG=T means clear prompt region first;
            L means new line; NIL means put it at the next char position)

    (ACE.MAC.CW.PROMPT.CLIP (PROGN (COND
                                    ((EQ FLG T)
                                    (DSPRESET ACE.CONTROL.WINDOW))
                                    ((EQ FLG (QUOTE L))
                                    (TERPRI ACE.CONTROL.WINDOW)))
                                (printout ACE.CONTROL.WINDOW STRING)
                                NIL])
```

(**ACE.CONFIRMIT**
```
  [LAMBDA (CONFIRMSTRING FLG WHICHKEYS?)                          (* PmT "25-Apr-85 17:47")

            (* Prints CONFIRMSTRING in A.C.W prompt region; then waits for the button form WHICHKEYS? to become true.
            WHICHKEYS? defaults to LEFT. Code identfies the valid button forms)
```

```
        (OR WHICHKEYS? (SETQ WHICHKEYS? (QUOTE LEFT)))
        (ACE.TELLEM CONFIRMSTRING FLG)
        (DISMISS 100 NIL T)
        (RESETFORM (CURSOR (SELECTQ WHICHKEYS?
                               (LEFT ACE.LEFTMOUSE.CURSOR)
                               (MIDDLE ACE.MIDDLEMOUSE.CURSOR)
                               (RIGHT ACE.RIGHTMOUSE.CURSOR)
                               (ANY (PROGN (SETQ WHICHKEYS? (QUOTE (NOT UP)))
                                           ACE.ALLMOUSE.CURSOR))
                               NIL))
             (do (GETMOUSESTATE) until (NEQ LASTMOUSEBUTTONS 0))
             (PROG1 (EVAL (MOUSESTATE-EXPR WHICHKEYS? T))
                 (do (GETMOUSESTATE) until (EQP LASTMOUSEBUTTONS 0)))])
```

(**ACE.DEFINE.SEQ.WINDOW**

```
  [LAMBDA NIL                                              (* PmT " 2-May-85 20:32")
    [COND
        ((ACE.CONFIRMIT "Click LEFT to Create a Sequence Window." T)
         (AND ACE.SEQ.WINDOW (CLOSEW ACE.SEQ.WINDOW))
         (SETQ ACE.SEQ.WINDOW (CREATEW (LIST 50 50 (IPLUS ACE.SEQ.WIDTH 8)
                                             (IPLUS ACE.SEQ.HEIGHT 17))
                                 (QUOTE "Animation Sequence Window")
                                 4))
         (ACE.TELLEM "Position the Sequence Window" T)
         (MOVEW ACE.SEQ.WINDOW))
        ((AND ACE.SEQ.WINDOW (ILEQ ACE.SEQ.WIDTH (WINDOWPROP ACE.SEQ.WINDOW (QUOTE WIDTH)))
              (ILEQ ACE.SEQ.HEIGHT (WINDOWPROP ACE.SEQ.WINDOW (QUOTE HEIGHT)))
              (ACE.CONFIRMIT "Click LEFT to Keep Current Window." T))
                                                         (* CLEARW ACE.SEQ.WINDOW)
         )
        (T (ACE.TELLEM "CAUTION:  Enter NIL if Unsure at this Stage." T)
           (SETQ ACE.SEQ.WINDOW (EVAL (ACE.ASKEM "Enter the Window: " (QUOTE L)
                                            NIL 120]
    (OR ACE.SEQ.WINDOW (ACE.DEFINE.SEQ.WINDOW])
```

(**ACE.FIGURE.OUT.WINDOW**

```
  [LAMBDA (REGION/POSITION)                                (* PmT "22-Apr-85 19:05")

          (* This is where all reasoning about which window to use and where offsets should be placed goes.
          Right now (4/20/85) Trillium's just gonna go with positions; but that should
          (?) change)

    (COND
        (ACE.RUNNING.UNDER.TRILLIUM (DISPLAY.FRAME CURRENT.FRAME)
            (ACE.CREATE.EDITING.BORDER (QUOTE INVERT))    (* KEEP OFFSETS THE SAME FOR NOW)
            )
        ((POSITIONP REGION/POSITION)
         (SETQ ACE.SEQ.WINDOW.XOFF (CAR REGION/POSITION))
         (SETQ ACE.SEQ.WINDOW.YOFF (CDR REGION/POSITION)))
        (T (SETQ ACE.SEQ.WINDOW.XOFF 0)
           (SETQ ACE.SEQ.WINDOW.YOFF 0)
           (ACE.DEFINE.SEQ.WINDOW)))
    (ACE.SET.SEQ.CLIP.REGION])
```

(**ACE.RETURN.CLOSEST.VERTEX**

```
  [LAMBDA (POINT REGION)                                   (* PmT "28-Nov-84 16:15")
    (PROG (NEW.XCOOR NEW.YCOOR)
          [COND
             [(IGREATERP (CAR POINT)
                  (SETQ NEW.XCOOR (fetch (REGION RIGHT) of REGION]
             [(ILESSP (CAR POINT)
                  (SETQ NEW.XCOOR (fetch (REGION LEFT) of REGION]
             (T (SETQ NEW.XCOOR (CAR POINT]
          [COND
             [(IGREATERP (CDR POINT)
                  (SETQ NEW.YCOOR (fetch (REGION TOP) of REGION]
             [(ILESSP (CDR POINT)
                  (SETQ NEW.YCOOR (fetch (REGION BOTTOM) of REGION]
             (T (SETQ NEW.YCOOR (CDR POINT]
          (RETURN (CONS NEW.XCOOR NEW.YCOOR)])
```

(**ACE.NEW.SEQ.ASST**

```
  [LAMBDA (FIXED MOVE DUM)                                 (* PmT "23-Jan-85 19:52")
    (COND
        ((NULL MOVE)
         (ACE.UPD.CONTROL.WINDOW (QUOTE CURSOR)
             FIXED)
         FIXED)
        (T [ACE.UPD.CONTROL.WINDOW (QUOTE CURSOR)
               (CONS (ABS (IDIFFERENCE (fetch (POSITION XCOORD) of MOVE)
                             (fetch (POSITION XCOORD) of FIXED)))
                     (ABS (IDIFFERENCE (fetch (POSITION YCOORD) of MOVE)
```

```
                                        (fetch (POSITION YCOORD) of FIXED]
            MOVE])
```

(**ACE.DELAY.FRAME.ASST**
```
  [LAMBDA (ITEM MENU MOUSE)                                   (* PmT "21-Dec-84 16:42")
    (COND
      [(FIXP (CADR ITEM))
       (ACE.QUICKDRAW&UPD (LDIFF ACE.CURRENT.SEQUENCE (CDR (NTH ACE.CURRENT.SEQUENCE (CADR ITEM]
      (T NIL])
```

(**ACE.SETUP.CW.CLIPPING.REGIONS**
```
  [LAMBDA (MENU)                                              (* PmT "23-Apr-85 13:47")

          (* Sets the clipping region on ACE.CONTROL.WINDOW; There is a menu region, prompt region and status region)

    (PROG (NORMAL ABOVEMENU INFO)
          (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE NORMAL.CLIP.REGION)
                 (SETQ NORMAL (DSPCLIPPINGREGION NIL ACE.CONTROL.WINDOW)))
          [WINDOWPROP ACE.CONTROL.WINDOW (QUOTE ABOVEMENU.CLIP.REGION)
                 (SETQ ABOVEMENU (CREATEREGION (fetch (REGION LEFT) of NORMAL)
                                       (fetch IMAGEHEIGHT of MENU)
                                       (fetch (REGION WIDTH) of NORMAL)
                                       (IDIFFERENCE (fetch (REGION HEIGHT) of NORMAL)
                                                (fetch IMAGEHEIGHT of MENU]
          [WINDOWPROP ACE.CONTROL.WINDOW (QUOTE INFO.CLIP.REGION)
                 (SETQ INFO (CREATEREGION (fetch (REGION LEFT) of ABOVEMENU)
                                   (fetch (REGION BOTTOM) of ABOVEMENU)
                                   130
                                   (IDIFFERENCE (fetch (REGION HEIGHT) of ABOVEMENU)
                                            5]
          (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE PROMPT.CLIP.REGION)
                 (CREATEREGION (IPLUS 3 (fetch (REGION RIGHT) of INFO))
                        (fetch (REGION BOTTOM) of ABOVEMENU)
                        (IDIFFERENCE (fetch (REGION RIGHT) of ABOVEMENU)
                               (IPLUS 3 (fetch (REGION RIGHT) of INFO)))
                        (IDIFFERENCE (fetch (REGION HEIGHT) of ABOVEMENU)
                               5)))
          (DRAWLINE (ADD1 (fetch (REGION RIGHT) of INFO))
                 (fetch (REGION BOTTOM) of ABOVEMENU)
                 (ADD1 (fetch (REGION RIGHT) of INFO))
                 (fetch (REGION TOP) of ABOVEMENU)
                 1
                 (QUOTE PAINT)
                 ACE.CONTROL.WINDOW)
          (DSPFILL (CREATEREGION 0 0 (fetch IMAGEWIDTH of MENU)
                          (fetch IMAGEHEIGHT of MENU))
                 38505
                 (QUOTE PAINT)
                 ACE.CONTROL.WINDOW])
```

(**ACE.CHECKSTUFF**
```
  [LAMBDA (CONDITIONS)                                        (* PmT "26-Oct-84 16:10")
    (COND
      ((EQ CONDITIONS (QUOTE SEQ))
       (OR ACE.CURRENT.SEQUENCE (ACE.TELLEM "No Current Sequence defined.  Aborted" T)))
      ((EQ CONDITIONS (QUOTE FRAME))
       (OR ACE.CURRENT.FRAME (ACE.TELLEM "No Current Frame.  Aborted" T)))
      ((EQ CONDITIONS (QUOTE TAIL))
       (OR ACE.FRAME.TAIL (ACE.TELLEM "Sequence is at End." T])
```

(**ACE.UPD.CONTROL.WINDOW**
```
  [LAMBDA (ITEM VALUE)                                        (* MD "18-Jun-85 16:19")

          (* This puts info in the status region of the control window; ITEM one of: CURSOR, FRAME, DEVICE, OPERATION, UPD,
          T, RESET. VALUE is the value for the ITEM; The ITEMs and VALUEs are stored as WINDOWPROPs on A.C.W)

    (ACE.MAC.CW.INFO.CLIP (COND
                            ((AND (KEYDOWNP (QUOTE T))
                                  (EQ ITEM (QUOTE CURSOR)))
                             (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
                                    VALUE))
                            ((EQ ITEM (QUOTE FRAME))
                             [COND
                               ((EQ VALUE T)
                                (SETQ VALUE (COND
                                              ((NULL ACE.CURRENT.SEQUENCE)
                                               (QUOTE NA))
                                              ((EQ ACE.CURRENT.SEQUENCE ACE.FRAME.TAIL)
                                               (QUOTE START))
                                              (T (LENGTH (LDIFF ACE.CURRENT.SEQUENCE ACE.FRAME.TAIL]
                             (ACE.UPD.CW.MULE (QUOTE ACE.FRAME)
                                    VALUE))
                            ((EQ ITEM (QUOTE DEVICE))
```

```
                                    (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE)
                                        VALUE))
                                ((EQ ITEM (QUOTE OPERATION))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION)
                                        VALUE))
                                ((EQ ITEM (QUOTE UPD))
                                  (ACE.UPD.CONTROL.WINDOW2 (QUOTE FRAME)
                                        T)
                                  (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION)
                                        (QUOTE OK))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
                                        (QUOTE NA)))
                                ((EQ ITEM T)
                                  (ACE.UPD.CW.MULE (QUOTE ACE.FRAME))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)))
                                ((EQ ITEM (QUOTE RESET))
                                  (ACE.UPD.CONTROL.WINDOW2 (QUOTE FRAME)
                                        T)
                                  (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE)
                                        (QUOTE MOUSE))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION)
                                        (QUOTE NA))
                                  (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
                                        (QUOTE NA])
```

(**ACE.UPD.CW.MULE**
```
  [LAMBDA (ITEM VALUE)                                           (* MD "18-Jun-85 16:49")

            (* An elaborate WINDOWPROPer. If VALUE is given, it's put on A.C.W as prop ITEM;
            if VALUE = NIL, returns the current value of ITEM. Also, writes the value in the status region of A.C.W.
            ITEM one of: ACE.CURSOR ACE.FRAME ACE.OPERATION ACE.DEVICE.
            Some restrictions of what VALUE can be (see code); Returns VALUE)

    (COND
       [(AND (KEYDOWNP (QUOTE T))
             (EQ ITEM (QUOTE ACE.CURSOR)))
        [COND
           ((OR (POSITIONP VALUE)
                (EQ VALUE (QUOTE NA)))
            (WINDOWPROP ACE.CONTROL.WINDOW ITEM VALUE))
           (T (SETQ VALUE (WINDOWPROP ACE.CONTROL.WINDOW ITEM]
        (ACE.UPD.CLEAR.SET.LINE 4)
        (COND
           ((POSITIONP VALUE)
            (printout ACE.CONTROL.WINDOW " Cursor: " (CAR VALUE)
                   " "
                   (CDR VALUE)
                   .SP 10))
           (T (printout ACE.CONTROL.WINDOW " Cursor: " VALUE .SP 10]
       ((EQ ITEM (QUOTE ACE.FRAME))
        [COND
           ((OR (FIXP VALUE)
                (EQ VALUE (QUOTE START))
                (EQ VALUE (QUOTE NA)))
            (WINDOWPROP ACE.CONTROL.WINDOW ITEM VALUE))
           (T (SETQ VALUE (WINDOWPROP ACE.CONTROL.WINDOW ITEM]
        (ACE.UPD.CLEAR.SET.LINE 1)
        (printout ACE.CONTROL.WINDOW " Frame: " VALUE .SP 20))
       ((EQ ITEM (QUOTE ACE.OPERATION))
        (COND
           ((NULL VALUE)
            (SETQ VALUE (WINDOWPROP ACE.CONTROL.WINDOW ITEM)))
           (T (WINDOWPROP ACE.CONTROL.WINDOW ITEM VALUE)))
        (ACE.UPD.CLEAR.SET.LINE 3)
        (printout ACE.CONTROL.WINDOW " State: " VALUE .SP 20))
       ((EQ ITEM (QUOTE ACE.DEVICE))
        [COND
           ((OR (EQ VALUE (QUOTE MOUSE))
                (EQ VALUE (QUOTE MM1201)))
            (WINDOWPROP ACE.CONTROL.WINDOW ITEM VALUE))
           (T (SETQ VALUE (WINDOWPROP ACE.CONTROL.WINDOW ITEM]
        (ACE.UPD.CLEAR.SET.LINE 2)
        (printout ACE.CONTROL.WINDOW " Device: " (COND
                                                   ((EQ VALUE (QUOTE MOUSE))
                                                    (QUOTE MOUSE))
                                                   ((EQ VALUE (QUOTE MM1201))
                                                    (QUOTE TABLET))
                                                   (T (QUOTE NA)))
                   .SP 10)))
       VALUE])
```

(**ACE.UPD.CLEAR.SET.LINE**

```
  [LAMBDA (LINES)                                                (* PmT "17-Dec-84 19:11")
    (MOVETOUPPERLEFT ACE.CONTROL.WINDOW)
    (RELMOVETO 0 (ITIMES (DSPLINEFEED NIL ACE.CONTROL.WINDOW)
                         (SUB1 LINES))
         ACE.CONTROL.WINDOW])
```

(ACE.CREATE.CONTROL.MENU
```
  [LAMBDA NIL                                                    (* MD "26-Jun-85 14:11")
    (SETQ ACE.CONTROL.MENU (create MENU
                                   ITEMS _ (QUOTE (("Get Sequence" (QUOTE I/O:GET)
                                                        "Fetch a sequence-file")
                                                   ("Edit Frame" (QUOTE EDIT)
                                                        "Edits the CURRENT frame")
                                                   ("Run Sequence" (QUOTE RUN)
                                                        "Runs the sequence"
                                                        (SUBITEMS ("Loop" (QUOTE LOOP)
                                                                      "Runs sequence repeatedly until you
                                                                      type a space")
                                                                  ("Loop part" (QUOTE SUBLOOP)
                                                                      "Runs part of the sequence repeatedly "))
                                                        )
                                                   ("Put Sequence" (QUOTE I/O:PUT)
                                                        "Writes current sequence out to a file")
                                                   ("New Frame" (QUOTE FRAME)
                                                        "Adds in another frame AFTER the current one")
                                                   ("Increment Frame" (QUOTE INC)
                                                        "Moves forward one frame and displays")
                                                   ("New Sequence" (QUOTE NEW)
                                                        "Make a new sequence from scratch")
                                                   ("Delete Frame" (QUOTE DEL)
                                                        "Removes CURRENT frame and smoothes over")
                                                   ("Decrement Frame" (QUOTE DEC)
                                                        "Goes back one frame")
                                                   ("Reset Sequence" (QUOTE RESET)
                                                        "Clears window and resets to start of sequence")
                                                   (" Adjust Timing Delays " (QUOTE TIME)
                                                        "Manipulate the timing adjustments")
                                                   ("Initialize MM1201 Tablet" (QUOTE INIT.MM1201)
                                                        "Sets up the Tablet for use")
                                                   ("Change compression %%" (QUOTE ACESETTHRESHOLD)
                                                        "Changes the space compression factor: 0 to 100 (100 =
                                                        max compression)")
                                                   ("Change Input Device" (QUOTE DEVICE)
                                                        "Select Mouse or Tablet (for now)")
                                                   ("Quit" (QUOTE QUIT)
                                                        "Exit ACE; Trillium user's MUST quit when done")))
                                   CENTERFLG _ T
                                   MENUCOLUMNS _ 3])
```

(ACE.SEQ.FETCH.WIDTH
```
  [LAMBDA NIL                                                    (* PmT "22-Apr-85 19:12")
    (FETCH (BITMAP BITMAPWIDTH) OF (FETCH (ACE.BLIT BITMAP) OF (CAR (FETCH (ACE.FRAME BLITS)
                                                                       OF (CAR ACE.CURRENT.SEQUENCE]
```

(ACE.SEQ.FETCH.HEIGHT
```
  [LAMBDA NIL                                                    (* PmT "22-Apr-85 19:14")
    (fetch (BITMAP BITMAPHEIGHT) of (fetch (ACE.BLIT BITMAP) of (CAR (fetch (ACE.FRAME BLITS) of (CAR
                                                                                                  ACE.CURRENT.SEQUENCE
                                                                                                  ])
```

(ACE.SET.SEQ.CLIP.REGION
```
  [LAMBDA (LEFT BOTTOM WIDTH HEIGHT)                             (* "PmT" "17-Apr-85 18:03")
    (OR LEFT (SETQ LEFT ACE.SEQ.WINDOW.XOFF))
    (OR BOTTOM (SETQ BOTTOM ACE.SEQ.WINDOW.YOFF))
    (OR WIDTH (SETQ WIDTH ACE.SEQ.WIDTH))
    (OR HEIGHT (SETQ HEIGHT ACE.SEQ.HEIGHT))
    (WINDOWPROP ACE.CONTROL.WINDOW (QUOTE SEQUENCE.CLIPPING.REGION)
         (CREATEREGION LEFT BOTTOM WIDTH HEIGHT])
```

(ACE.ASKEM2
```
  [LAMBDA (STRING FLG DEFAULTANSWER TIMELIMIT? SPACES?)          (* PmT "22-Apr-85 15:56")
                                                                (* Like ASKEM but uses whole control window
                                                                (use cautiously))

    (OR TIMELIMIT? (SETQ TIMELIMIT? 60))
    (COND
       ((EQ FLG T)
        (DSPRESET ACE.CONTROL.WINDOW))
       ((EQ FLG (QUOTE L))
        (TERPRI ACE.CONTROL.WINDOW)))
    (MKATOM (PROMPTFORWORD STRING DEFAULTANSWER NIL ACE.CONTROL.WINDOW NIL TIMELIMIT?
                  (AND SPACES? (CHARCODE (EOL ESCAPE LF])
```

(**ACE.TELLEM2**
```
  [LAMBDA (STRING FLG)                                          (* PmT "19-Dec-84 19:15")
    (COND
        ((EQ FLG T)
         (DSPRESET ACE.CONTROL.WINDOW))
        ((EQ FLG (QUOTE L))
         (TERPRI ACE.CONTROL.WINDOW)))
    (printout ACE.CONTROL.WINDOW STRING)
    NIL])
```

(**ACE.UPD.CONTROL.WINDOW2**
```
  [LAMBDA (ITEM VALUE)                                          (* PmT "19-Dec-84 15:49")
    (COND
        ((EQ ITEM (QUOTE FRAME))
         [COND
             ((EQ VALUE T)
              (SETQ VALUE (COND
                              ((NULL ACE.CURRENT.SEQUENCE)
                               (QUOTE NA))
                              ((EQ ACE.CURRENT.SEQUENCE ACE.FRAME.TAIL)
                               (QUOTE START))
                              (T (LENGTH (LDIFF ACE.CURRENT.SEQUENCE ACE.FRAME.TAIL]
         (ACE.UPD.CW.MULE (QUOTE ACE.FRAME)
                VALUE))
        ((EQ ITEM (QUOTE DEVICE))
         (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE)
                VALUE))
        ((EQ ITEM (QUOTE OPERATION))
         (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION)
                VALUE))
        ((EQ ITEM (QUOTE CURSOR))
         (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
                VALUE))
        ((EQ ITEM (QUOTE UPD))
         (ACE.UPD.CONTROL.WINDOW2 (QUOTE FRAME)
                T)
         (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE))
         (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION)
                (QUOTE OK))
         (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
                (QUOTE NA)))
        ((EQ ITEM T)
         (ACE.UPD.CW.MULE (QUOTE ACE.FRAME))
         (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE))
         (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION))
         (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)))
        ((EQ ITEM (QUOTE RESET))
         (ACE.UPD.CONTROL.WINDOW2 (QUOTE FRAME)
                T)
         (ACE.UPD.CW.MULE (QUOTE ACE.DEVICE)
                (QUOTE MOUSE))
         (ACE.UPD.CW.MULE (QUOTE ACE.OPERATION)
                (QUOTE NA))
         (ACE.UPD.CW.MULE (QUOTE ACE.CURSOR)
                (QUOTE NA])
)
```

          (* * The following Macros set up restricting clipping regions)

```
(DECLARE: EVAL@COMPILE

(PUTPROPS ACE.MAC.CW.INFO.CLIP MACRO ((FORM)
                                         (RESETLST
                                             [RESETSAVE (PROGN (DSPCLIPPINGREGION (WINDOWPROP ACE.CONTROL.WINDOW
                                                                                       (QUOTE INFO.CLIP.REGION))
                                                                        ACE.CONTROL.WINDOW)
                                                               (DSPLEFTMARGIN (fetch (REGION LEFT)
                                                                                  of (WINDOWPROP ACE.CONTROL.WINDOW
                                                                                          (QUOTE INFO.CLIP.REGION
                                                                                              )))
                                                                        ACE.CONTROL.WINDOW))
                                                        (QUOTE (PROGN (DSPCLIPPINGREGION (WINDOWPROP
                                                                                       ACE.CONTROL.WINDOW
                                                                                       (QUOTE
                                                                                           NORMAL.CLIP.REGION
                                                                                           ))
                                                                             ACE.CONTROL.WINDOW)
                                                                      (DSPLEFTMARGIN (fetch (REGION LEFT)
                                                                                         of
                                                                                         (WINDOWPROP
                                                                                             ACE.CONTROL.WINDOW
                                                                                             (QUOTE
                                                                                                 NORMAL.CLIP.REGION
```

```
                                                                                                      )))
                                                                       ACE.CONTROL.WINDOW]
                                            FORM)))
(PUTPROPS ACE.MAC.CW.PROMPT.CLIP MACRO ((FORM)
                                        (RESETLST
                                           [RESETSAVE (PROGN (DSPCLIPPINGREGION (WINDOWPROP
                                                                                  ACE.CONTROL.WINDOW
                                                                                  (QUOTE
                                                                                    PROMPT.CLIP.REGION
                                                                                    ))
                                                              ACE.CONTROL.WINDOW)
                                                          (DSPLEFTMARGIN (fetch (REGION LEFT)
                                                                             of (WINDOWPROP
                                                                                  ACE.CONTROL.WINDOW
                                                                                  (QUOTE
                                                                                    PROMPT.CLIP.REGION
                                                                                    )))
                                                              ACE.CONTROL.WINDOW))
                                                       (QUOTE (PROGN (DSPCLIPPINGREGION (WINDOWPROP
                                                                                          ACE.CONTROL.WINDOW
                                                                                          (QUOTE
                                                                                            NORMAL.CLIP.REGION
                                                                                            ))
                                                                      ACE.CONTROL.WINDOW)
                                                                  (DSPLEFTMARGIN (fetch (REGION LEFT)
                                                                                     of
                                                                                     (WINDOWPROP
                                                                                       ACE.CONTROL.WINDOW
                                                                                       (QUOTE
                                                                                         NORMAL.CLIP.REGION
                                                                                         )))
                                                                      ACE.CONTROL.WINDOW]
                                            FORM)))
(PUTPROPS ACE.MAC.SEQ.CLIP MACRO ((FORM)
                                  (COND
                                    ((WINDOWPROP ACE.CONTROL.WINDOW (QUOTE SEQUENCE.CLIPPING.REGION))
                                     (RESETLST
                                        (RESETSAVE (DSPCLIPPINGREGION (WINDOWPROP ACE.CONTROL.WINDOW
                                                                        (QUOTE SEQUENCE.CLIPPING.REGION)
                                                                        )
                                                      ACE.SEQ.WINDOW)
                                           (LIST (QUOTE DSPCLIPPINGREGION)
                                                 (DSPCLIPPINGREGION NIL ACE.SEQ.WINDOW)
                                                 ACE.SEQ.WINDOW))
                                        FORM))
                                    (T FORM))))
)

(DECLARE: EVAL@COMPILE

(PUTPROPS ACE.MAC.FETCH.WIDTH MACRO [NIL (fetch (BITMAP BITMAPWIDTH) of (fetch (ACE.BLIT BITMAP)
                                                                          of (CAR (fetch (ACE.FRAME BLITS)
                                                                                     of (CAR
                                                                                          ACE.CURRENT.SEQUENCE
                                                                                          ])

(PUTPROPS ACE.MAC.FETCH.HEIGHT MACRO [NIL (fetch (BITMAP BITMAPHEIGHT) of (fetch (ACE.BLIT BITMAP)
                                                                            of (CAR (fetch (ACE.FRAME BLITS)
                                                                                       of (CAR
                                                                                            ACE.CURRENT.SEQUENCE
                                                                                            ])
)

(RPAQ ACE.LEFTMOUSE.CURSOR (CURSORCREATE (READBITMAP)
                                         8 8))


(16 16 "GOOL" "D@@D" "ELID" "ELID" "ELID" "ELID" "ELID" "ELID" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D"
    "GOOL")

(RPAQ ACE.MIDDLEMOUSE.CURSOR (CURSORCREATE (READBITMAP)
                                           8 8))


(16 16 "GOOL" "D@@D" "ECID" "ECID" "ECID" "ECID" "ECID" "ECID" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D"
    "GOOL")

(RPAQ ACE.RIGHTMOUSE.CURSOR (CURSORCREATE (READBITMAP)
                                          8 8))


(16 16 "GOOL" "D@@D" "EBGD" "EBGD" "EBGD" "EBGD" "EBGD" "EBGD" "EBGD" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D"
    "GOOL")

(RPAQ ACE.ALLMOUSE.CURSOR (CURSORCREATE (READBITMAP)
```

                                  8 8))

(16 16 "GOOL" "D@@D" "DMKD" "DMKD" "DMKD" "DMKD" "DMKD" "DMKD" "DMKD" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D" "D@@D"
     "GOOL")

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS ACE.CONTROL.WINDOW ACE.DIRECTORY ACE.SEQ.WINDOW ACE.SEQ.WIDTH ACE.SEQ.HEIGHT ACE.SEQ.WINDOW.XOFF
        ACE.SEQ.WINDOW.YOFF ACE.CURRENT.SEQUENCE ACE.CURRENT.SEQUENCE.NAME ACE.FRAME.TAIL ACE.CURRENT.FRAME
        ACE.VERTICAL.BLOCK ACE.AREA.THRESHOLD ACE.RUNNING.UNDER.TRILLIUM ACE.LEFTMOUSE.CURSOR
        ACE.MIDDLEMOUSE.CURSOR ACE.RIGHTMOUSE.CURSOR ACE.ALLMOUSE.CURSOR)
)

                (* * MENUS IN MAIN)

(DECLARE: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS ACE.CONTROL.MENU ACE.DELAY.MENU ACE.SET.DEVICE.MENU)
)

(SETQ ACE.CONTROL.WINDOW NIL)

(SETQ ACE.CONTROL.MENU NIL)

(SETQ ACE.DELAY.MENU NIL)

(SETQ ACE.SET.DEVICE.MENU NIL)

(DECLARE: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR **NLAMA**)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA**)
)

FUNCTION INDEX

MACRO INDEX

VARIABLE INDEX