```
    File created:   20-Apr-88 05:11:42  {DSK}<LISPFILES>DANCEROBJ.;15

      changes to:   (VARS DANCEROBJCOMS)
                    (FNS DANCEROBJ.GETFN2 DANCEROBJ.PUTFN DANCEROBJ.INIT DANCEROBJ.DISPLAY-AUX-LINES
                         DANCEROBJ.DISPLAYFN DANCEROBJ.EDIT DANCEROBJ.CREATEBUTTONFN DANCEROBJ.MAKEFMDESC DANCER-Y
                         DANCER-X DANCEROBJ.CREATE DANCEROBJ.MAKEMENU DANCEROBJ.Y-SCALED-CURVE DANCEROBJ.X-SCALED-POLY
                         DANCEROBJ.BOTH-SCALED-POLY DANCEROBJ.COMPUTE-AUXLINE-EXTENSIONS DANCEROBJ.X-SCALED-CURVE
                         DANCEROBJ.BOTH-SCALED-CURVE)

   previous date:   14-Apr-88 02:46:03  {DSK}<LISPFILES>DANCEROBJ.;10

     Read Table:    INTERLISP

        Package:    INTERLISP

          Format:    XCCS
```

```
(RPAQQ DANCEROBJCOMS
       ((FILES FREEMENU)
        (FNS DANCERMENU.CLOSEFN DANCEROBJ.CREATE DANCEROBJ.DISPLAYFN DANCEROBJ.IMAGEBOXFN DANCEROBJ.GETFN
             DANCEROBJ.GETFN2 DANCEROBJ.PUTFN DANCEROBJ.BUTTONEVENTINFN DANCEROBJ.COPYFN DANCEROBJ.INIT
             DANCEROBJ.INITFORMATIONINFO)
        (FNS DANCEROBJ.MAKEMENU DANCEROBJ.EDIT DANCEROBJ.MAKETERMTABLE DANCEROBJ.CREATEBUTTONFN
             DANCEROBJ.FACEBUTTONFN DANCEROBJ.MAKEFMDESC DANCEROBJ.SORTLOCS DANCEROBJ.ROT90 DANCEROBJ.COMPOUND)
        (COMS                                                          ; Auxiliary Line functions
              (FNS DANCER-Y DANCER-X DANCEROBJ.COMPUTE-AUXLINE-EXTENSIONS IAVG)
              (FNS DANCEROBJ.X-SCALED-CURVE DANCEROBJ.Y-SCALED-CURVE DANCEROBJ.BOTH-SCALED-CURVE
                   DANCEROBJ.DISPLAY-AUX-LINES)
              (FNS DANCEROBJ.X-SCALED-POLY DANCEROBJ.BOTH-SCALED-POLY)

              ;; Knots for semi-circles (actually 30% ellipses)

              (VARS DANCEROBJ.TOP-KNOTS DANCEROBJ.BOT-KNOTS DANCEROBJ.L-KNOTS DANCEROBJ.R-KNOTS)

              ;; 1/4 circle & ellipse quadrant knots

              (VARS DANCEROBJ.LU-KNOTS DANCEROBJ.LB-KNOTS DANCEROBJ.RU-KNOTS DANCEROBJ.RB-KNOTS)
              (RECORDS DANCER-AUX-LINE))
        (VARS (*DANCER-FONT-SIZE* 12)
              (DANCEROBJ.MENU)
              (DANCEROBJ.FORMATIONINFO)
              (DANCEROBJ.FMDESC)
              (DANCEROBJ.TERMTABLE (DANCEROBJ.MAKETERMTABLE))
              DANCEROBJ.INITIAL.FORMATION.INFO)
        (ADDVARS (IMAGEOBJGETFNS (DANCEROBJ.GETFN))
               (IMAGEOBJGETFNS (DANCEROBJ.GETFN2)))
        (DECLARE%: DONTEVAL@LOAD DOCOPY (P (DANCEROBJ.INIT)))
        (RECORDS FORMATION)))

(FILESLOAD FREEMENU)

(DEFINEQ

(DANCERMENU.CLOSEFN
  [LAMBDA (W)                                                          (* jds "22-Apr-86 16:30")

            (* CLOSE a TEdit menu window%: Detach the menu, then reshape the remaining windows to take up the remaining space)

    (PROG ((MAINW (WINDOWPROP W 'MAINWINDOW))
           TEXTOBJ HEIGHT OHEIGHT OBOTTOM WBOTTOM WINDOWS)
          (DETACHWINDOW W)                                             (* So detach this window.)
          (COND
             ((IGREATERP (FLENGTH (ATTACHEDWINDOWS MAINW))
                     1)
              [SETQ OHEIGHT (fetch HEIGHT of (WINDOWPROP W 'REGION]
              [SETQ OBOTTOM (fetch BOTTOM of (WINDOWPROP W 'REGION]
              (CLOSEW W)
              [SETQ WINDOWS (SORT (ATTACHEDWINDOWS MAINW)
                                  (FUNCTION (LAMBDA (WW)
                                              (fetch BOTTOM of (WINDOWPROP WW 'REGION]
              (for WW in WINDOWS when (IGEQ [SETQ WBOTTOM (fetch BOTTOM of (WINDOWPROP WW 'REGION]
                                            OBOTTOM)
                 do (MOVEW WW (fetch LEFT of (WINDOWPROP WW 'REGION))
                           (IDIFFERENCE WBOTTOM OHEIGHT])

(DANCEROBJ.CREATE
  [LAMBDA (FORMATION FACINGDIRS IDS LINES)                             ; Edited 16-Apr-88 18:18 by

    ;; RETURNS an image object representing the square dance formation FORMATION, with the dancers facing in FACINGDIRS, and labelled with
    ;; IDS.

    ;; ALSO with auxiliary lines described in LINES.

    (PROG ((OBJ (IMAGEOBJCREATE NIL DANCERIMAGEFNS)))
          (IMAGEOBJPROP OBJ 'FORMATION FORMATION)
          (IMAGEOBJPROP OBJ 'FACING FACINGDIRS)
          (IMAGEOBJPROP OBJ 'IDS IDS)
          (IMAGEOBJPROP OBJ 'LINES LINES)
          (RETURN OBJ])

(DANCEROBJ.DISPLAYFN
  [LAMBDA (IMAGEOBJ IMAGE.STREAM)                                    ; Edited 20-Apr-88 00:10 by

    ;; Display a dancer diagram on some image stream

    (LET* ((AUX-LINE-OFFSETS (DANCEROBJ.COMPUTE-AUXLINE-EXTENSIONS IMAGEOBJ IMAGE.STREAM))
           (CURX (+ (CAR AUX-LINE-OFFSETS)
                    (DSPXPOSITION NIL IMAGE.STREAM)))
           (CURY (+ (CADR AUX-LINE-OFFSETS)
                    (DSPYPOSITION NIL IMAGE.STREAM)))
           (FONT (FONTCREATE 'DANCER *DANCER-FONT-SIZE* NIL NIL IMAGE.STREAM))
           (FONTDESCENT (FONTPROP FONT 'DESCENT))
           [SIZE (IMAX (CHARWIDTH (CHARCODE a)
                                  FONT)
                       (FONTPROP FONT 'HEIGHT]
           (FORMATION (IMAGEOBJPROP IMAGEOBJ 'FORMATION))
           (FINFO (LISTGET DANCEROBJ.FORMATIONINFO FORMATION))
           (WIDTH (FIXR (FTIMES (fetch (FORMATION FWIDTH) of FINFO)
                                SIZE)))
           (HEIGHT (FIXR (FTIMES (fetch (FORMATION FHEIGHT) of FINFO)
                                 SIZE)))
           (DANCER-LOCATIONS (fetch (FORMATION FLOCS) of FINFO))
           (DANCER-FACING-DIRECTIONS (IMAGEOBJPROP IMAGEOBJ 'FACING))
           (AUXILIARY-LINES (IMAGEOBJPROP IMAGEOBJ 'LINES))
           (DANCER-IDS (IMAGEOBJPROP IMAGEOBJ 'IDS))
           (IMAGEBOX (OR (IMAGEOBJPROP IMAGEOBJ 'BOUNDBOX)
                         (DANCEROBJ.IMAGEBOXFN IMAGEOBJ IMAGE.STREAM)))
           (YDESC (IMINUS (fetch (IMAGEBOX YDESC) of IMAGEBOX)))
           (OFONT (DSPFONT NIL IMAGE.STREAM))
           ID FACING)
          (RESETLST
              (RESETSAVE (SETTERMTABLE DANCEROBJ.TERMTABLE))
              [RESETSAVE (DSPFONT FONT IMAGE.STREAM)
                     '(AND (DSPFONT OLDVALUE IMAGE.STREAM]
              [for SPOT in DANCER-LOCATIONS do
                                                ;; Run thru the spots in the formation, drawing each one in turn.  Run thru every spot even if
                                                ;; facing or ID info hasn't been provided for it.

                                                     ;; (These can't be AS's, since one may run out before the other):

                                                     (SETQ FACING (pop DANCER-FACING-DIRECTIONS))
                                                     (SETQ ID (pop DANCER-IDS))

                                                     ;; Print the dancer ID:

                                                     (MOVETO (IPLUS CURX (FIXR (FTIMES (fetch XCOORD of SPOT)
                                                                                       SIZE)))
                                                            (IPLUS CURY YDESC FONTDESCENT
                                                                   (FIXR (FTIMES (fetch YCOORD of SPOT)
                                                                                 SIZE)))
                                                            IMAGE.STREAM)
                                                     (AND ID (PRIN1 ID IMAGE.STREAM))

                                                ;; Now print the dancer image (box or circle).  The MOVETO is there because INTERPRESS
                                                ;; streams do the wrong thing with 0-wide chars sometimes:

                                                     (MOVETO (IPLUS CURX (FIXR (FTIMES (fetch XCOORD of SPOT)
                                                                                       SIZE)))
                                                            (IPLUS CURY YDESC FONTDESCENT
                                                                   (FIXR (FTIMES (fetch YCOORD of SPOT)
                                                                                 SIZE)))
                                                            IMAGE.STREAM)
                                                     (\OUTCHAR IMAGE.STREAM (SELECTQ (U-CASE FACING)
                                                                                (N (CHARCODE e))
                                                                                (S (CHARCODE c))
                                                                                (E (CHARCODE b))
                                                                                (W (CHARCODE d))
                                                                                (NS (CHARCODE g))
                                                                                (EW (CHARCODE f))
                                                                                ((P PH)
                                                                                     (CHARCODE h))
                                                                                ((-- NONE)
                                                                                     (CHARCODE a))
                                                                                ((R R-- RNONE)
                                                                                     (CHARCODE o))
                                                                                (RE (CHARCODE p))
                                                                                (RW (CHARCODE r))
                                                                                (RN (CHARCODE s))
                                                                                (RS (CHARCODE q))
                                                                                (REW (CHARCODE t))
                                                                                (RNS (CHARCODE u))
                                                                                (RNW (CHARCODE v))
                                                                                (RNE (CHARCODE w))
                                                                                (RSE (CHARCODE x))
                                                                                (RSW (CHARCODE y))
                                                                                (RNNE (CHARCODE ^A))
```

```
                                                                        (RENE (CHARCODE ^B))
                                                                        (RESE (CHARCODE ^C))
                                                                        (RSSE (CHARCODE ^D))
                                                                        (RSSW (CHARCODE ^E))
                                                                        (RWSW (CHARCODE ^F))
                                                                        (RWNW (CHARCODE ^G))
                                                                        (RNNW (CHARCODE ^H))
                                                                        ((X BLANK XX SP)
                                                                            (CHARCODE m))
                                                                        (CHARCODE a]
                  (DANCEROBJ.DISPLAY-AUX-LINES AUXILIARY-LINES IMAGE.STREAM DANCER-LOCATIONS SIZE CURX CURY
                        FONTDESCENT YDESC))]])
```

(DANCEROBJ.IMAGEBOXFN

```
  [LAMBDA (IMAGEOBJ IMAGE.STREAM CURRENT.X RIGHT.MARGIN)          ; Edited 14-Apr-88 02:19 by

    ;; returns an imagebox describing the size of the dancer diagram

    (PROG* ((AUX-LINE-OFFSETS (DANCEROBJ.COMPUTE-AUXLINE-EXTENSIONS IMAGEOBJ IMAGE.STREAM))
            (FONT (FONTCREATE 'DANCER *DANCER-FONT-SIZE* NIL NIL IMAGE.STREAM))
            [SIZE (IMAX (CHARWIDTH 97 FONT)
                        (FONTPROP FONT 'HEIGHT]
            (FORMATION (IMAGEOBJPROP IMAGEOBJ 'FORMATION))
            (FINFO (LISTGET DANCEROBJ.FORMATIONINFO FORMATION))
            (WIDTH (+ (FIXR (FTIMES (fetch (FORMATION FWIDTH) of FINFO)
                                    SIZE))
                      (CAR AUX-LINE-OFFSETS)
                      (CADDR AUX-LINE-OFFSETS)))
            (HEIGHT (+ (FIXR (FTIMES (fetch (FORMATION FHEIGHT) of FINFO)
                                     SIZE))
                       (CADR AUX-LINE-OFFSETS)
                       (CADDDR AUX-LINE-OFFSETS)))
            (LINES (IMAGEOBJPROP IMAGEOBJ 'LINES))
           VJUST HJUST)
           (RETURN (create IMAGEBOX
                          XSIZE _ WIDTH
                          YSIZE _ HEIGHT
                          YDESC _ (SELECTQ VJUST
                                      (TOP 0)
                                      (BOT HEIGHT)
                                      (LRSH (IDIFFERENCE HEIGHT SIZE)
                                            1))
                          XKERN _ (SELECTQ HJUST
                                      (RIGHT WIDTH)
                                      (CENTER (LRSH WIDTH 1))
                                      0])
```

(DANCEROBJ.GETFN

```
  [LAMBDA (STREAM)                                                 (* jds " 8-Dec-85 16:50")

              (* * Reads an imageobject that describes a formation of dancers)

    (PROG* ((FORMATION (\ATMIN STREAM))
            (FACINGDIRECTIONS (\ARBIN STREAM))
            (IDENTITIES (\ARBIN STREAM)))
           (RETURN (DANCEROBJ.CREATE FORMATION FACINGDIRECTIONS IDENTITIES])
```

(DANCEROBJ.GETFN2

```
  [LAMBDA (STREAM)                                                 ; Edited 20-Apr-88 02:52 by

;;; Reads an imageobject that describes a formation of dancers

    (PROG* ((FORMATION (\ATMIN STREAM))
            (FACINGDIRECTIONS (\ARBIN STREAM))
            (IDENTITIES (\ARBIN STREAM))
            (LINES (\ARBIN STREAM)))
           (RETURN (DANCEROBJ.CREATE FORMATION FACINGDIRECTIONS IDENTITIES LINES])
```

(DANCEROBJ.PUTFN

```
  [LAMBDA (BMOBJ STREAM)                                           ; Edited 20-Apr-88 02:51 by

;;; Put a description of a group of dancers into a file

    (PROG* [(FORMATION (IMAGEOBJPROP BMOBJ 'FORMATION))
            (FACINGDIRECTIONS (IMAGEOBJPROP BMOBJ 'FACING))
            (IDENTITIES (IMAGEOBJPROP BMOBJ 'IDS))
            (LINES (IMAGEOBJPROP BMOBJ 'LINES]
           (\ATMOUT STREAM FORMATION)
           (\ARBOUT STREAM FACINGDIRECTIONS)
           (\ARBOUT STREAM IDENTITIES)
           (\ARBOUT STREAM LINES])
```

(DANCEROBJ.BUTTONEVENTINFN

```
  [LAMBDA (IMAGEOBJ WINDOW)                                        (* jds "18-Dec-85 18:17")

            (* * the user has pressed a button inside the DANCER object IMAGEOBJ.
            Bring up a menu of DANCER edit operations.)

    (PROG* NIL
           (COND
              ([MENU (OR DANCEROBJ.MENU (SETQ DANCEROBJ.MENU (create MENU
                                                                  ITEMS _ '(Edit% Dancers]
               (DANCEROBJ.EDIT IMAGEOBJ)
               (RETURN 'CHANGED])
```

(DANCEROBJ.COPYFN

```
  [LAMBDA (IMAGEOBJ)                                               (* jds "18-Dec-85 15:04")

            (* RETURNS an image object representing the square dance formation FORMATION, with the dancers facing in
            FACINGDIRS, and labelled with IDS.)

    (PROG ((OBJ (IMAGEOBJCREATE NIL DANCERIMAGEFNS)))
          (IMAGEOBJPROP OBJ 'FORMATION (IMAGEOBJPROP IMAGEOBJ 'FORMATION))
          (IMAGEOBJPROP OBJ 'FACING (IMAGEOBJPROP IMAGEOBJ 'FACING))
          (IMAGEOBJPROP OBJ 'IDS (IMAGEOBJPROP IMAGEOBJ 'IDS))
          (RETURN OBJ])
```

(DANCEROBJ.INIT

```
  [LAMBDA NIL                                                      ; Edited 20-Apr-88 02:52 by

    ;; Initialization for the DANCEROBJ imagefns vector.

    (SETQ DANCERIMAGEFNS (IMAGEFNSCREATE (FUNCTION DANCEROBJ.DISPLAYFN)
                                (FUNCTION DANCEROBJ.IMAGEBOXFN)
                                (FUNCTION DANCEROBJ.PUTFN)
                                (FUNCTION DANCEROBJ.GETFN2)
                                (FUNCTION DANCEROBJ.COPYFN)
                                (FUNCTION DANCEROBJ.BUTTONEVENTINFN)
                                (FUNCTION NILL)
                                (FUNCTION NILL)
                                (FUNCTION NILL)
                                (FUNCTION NILL)
                                (FUNCTION NILL)
                                (FUNCTION NILL)
                                (FUNCTION NILL)
                                'DANCEROBJ))
    (SETQ DANCEROBJ.FORMATIONINFO NIL)
    (DANCEROBJ.INITFORMATIONINFO DANCEROBJ.INITIAL.FORMATION.INFO)
                                                                   ; Set up the intial formation info from the intial specs
    (SETQ DANCEROBJ.FMDESC (DANCEROBJ.MAKEFMDESC])
```

(DANCEROBJ.INITFORMATIONINFO

```
  [LAMBDA (FORMATIONSPECS)                                         (* jds "30-Dec-85 16:23")

            (* * Given a list of formation specs (as a PLIST)%, expand the definitions, and add them to
            DANCEROBJ.FORMATIONINFO)

    (for NAME in FORMATIONSPECS by (CDDR NAME) as DESC in (CDR FORMATIONSPECS) by (CDDR DESC)
       do (SETQ DANCEROBJ.FORMATIONINFO (NCONC DANCEROBJ.FORMATIONINFO (LIST NAME (DANCEROBJ.COMPOUND DESC])
)

(DEFINEQ
```

(DANCEROBJ.MAKEMENU

```
  [LAMBDA (MAINWINDOW)                                             ; Edited 16-Apr-88 18:11 by

;;; Create a free menu to be used in creating and editing dancer objects

    (PROG [(MAINW (OR MAINWINDOW (WHICHW]
          (DANCEROBJ.MAKEFMDESC)
          (SETQ DANCEROBJ.FREEMENU (FREEMENU DANCEROBJ.FMDESC "Dancer Diagram Menu"))
          (WINDOWPROP DANCEROBJ.FREEMENU 'MAINWINDOW MAINW)
          (WINDOWADDPROP DANCEROBJ.FREEMENU 'CLOSEFN (FUNCTION DANCERMENU.CLOSEFN))
          (ATTACHWINDOW DANCEROBJ.FREEMENU (OR MAINW (WHICHW))
                 'TOP
                 'JUSTIFY
                 'LOCALCLOSE])
```

(DANCEROBJ.EDIT

```
  [LAMBDA (OBJ)                                                    ; Edited 20-Apr-88 00:31 by

;;; Create a free menu to be used in creating and editing dancer objects

    (PROG* ((FORMATION (IMAGEOBJPROP OBJ 'FORMATION))
            (IDS (IMAGEOBJPROP OBJ 'IDS))
            (FACING (IMAGEOBJPROP OBJ 'FACING))
            (LINES (IMAGEOBJPROP OBJ 'LINES))
            (DONEITEM (FM.GETITEM 'FINISHER NIL DANCEROBJ.FREEMENU))
            (FORMATIONITEM (FM.GETITEM 'FORMATION NIL DANCEROBJ.FREEMENU))
            (IDITEM (FM.GETITEM 'IDENT NIL DANCEROBJ.FREEMENU))
            (FACINGITEM (FM.GETITEM 'FACE NIL DANCEROBJ.FREEMENU))
            [IDSTRING (CONCATLIST (for ID in IDS join (LIST ID " "]
            [FACESTRING (CONCATLIST (for FACE in FACING join (LIST FACE " "]
            MENUSTATE)
           (while (LISTGET (FM.GETSTATE DANCEROBJ.FREEMENU)
                          'FINISHER)
              do                                                ; Make sure the DONT toggle in the menu is turned off.
                 (FM.CHANGESTATE DONEITEM NIL DANCEROBJ.FREEMENU))
           (FM.CHANGESTATE FORMATIONITEM FORMATION DANCEROBJ.FREEMENU)
                                                                ; Fill in the formation, ID's, and facing directions in the menu from
                                                                ; this object
           (FM.CHANGESTATE IDITEM IDSTRING DANCEROBJ.FREEMENU)
           (FM.CHANGESTATE FACINGITEM FACESTRING DANCEROBJ.FREEMENU)
           (for LINE in (APPEND LINES '("" "" "" "" "" "")) as ID
              in '(Aux1 Aux2 Aux3 Aux4 Aux5 Aux6 Aux7 Aux8) do (FM.CHANGESTATE (FM.GETITEM ID NIL
                                                                                           DANCEROBJ.FREEMENU)
                                                                       (MKSTRING LINE)
                                                                       DANCEROBJ.FREEMENU))
           (SPAWN.MOUSE)                                        ; Since we're the mouse process, need another one to fiddle the
                                                                ; menu while we're waiting.
           (until (LISTGET (FM.GETSTATE DANCEROBJ.FREEMENU)
                         'FINISHER)
              do                                                ; Wait until the user claims to be finished in the menu
                 (DISMISS 250))
           (SETQ MENUSTATE (FM.GETSTATE DANCEROBJ.FREEMENU))    ; Find out what he did
           (FM.CHANGESTATE DONEITEM NIL DANCEROBJ.FREEMENU)
           (IMAGEOBJPROP OBJ 'FORMATION (LISTGET MENUSTATE 'FORMATION))
                                                                ; Fill in the object's declarations from the menu's changes
           [IMAGEOBJPROP OBJ 'IDS (READFILE (OPENSTRINGSTREAM (LISTGET MENUSTATE 'IDENT]
           [IMAGEOBJPROP OBJ 'FACING (READFILE (OPENSTRINGSTREAM (LISTGET MENUSTATE 'FACE]
           [IMAGEOBJPROP OBJ 'LINES (READFILE (OPENSTRINGSTREAM (CONCAT (LISTGET MENUSTATE 'Aux1)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux2)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux3)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux4)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux5)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux6)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux7)
                                                                      " "
                                                                      (LISTGET MENUSTATE 'Aux8]
           (IMAGEOBJPROP OBJ 'BITMAP NIL)                       ; And invalidate the image cache so it gets redisplayed

       ])
```

(DANCEROBJ.MAKETERMTABLE

```
  [LAMBDA NIL
    (LET ((TTBL (COPYTERMTABLE \ORIGTERMTABLE)))
         (for I from 1 to 255 do (ECHOCHAR I 'REAL TTBL))
         TTBL])
```

(DANCEROBJ.CREATEBUTTONFN

```
  [LAMBDA (ITEM MENU BUTTON)                                    ; Edited 20-Apr-88 00:27 by
```

;;; He hit the CREATE button. Create him an object.

```
    (PROG* [(MENUSTATE (FM.GETSTATE MENU))
            (FORMATION (LISTGET MENUSTATE 'FORMATION))
            (IDSTREAM (OPENSTRINGSTREAM (LISTGET MENUSTATE 'IDENT)
                             'INPUT))
            (IDS (READFILE IDSTREAM))
            (FACESTREAM (OPENSTRINGSTREAM (LISTGET MENUSTATE 'FACE)
                               'INPUT))
            (LINESTREAM (OPENSTRINGSTREAM (CONCAT (LISTGET MENUSTATE 'Aux1)
                                                  " "
                                                  (LISTGET MENUSTATE 'Aux2)
                                                  " "
                                                  (LISTGET MENUSTATE 'Aux3)
                                                  " "
                                                  (LISTGET MENUSTATE 'Aux4)
                                                  " "
                                                  (LISTGET MENUSTATE 'Aux5)
                                                  " "
                                                  (LISTGET MENUSTATE 'Aux6)
                                                  " "
                                                  (LISTGET MENUSTATE 'Aux7)
                                                  " "
```

```
                                                     (LISTGET MENUSTATE 'Aux8))
                              'INPUT))
            (LINES (READFILE LINESTREAM))
            (FACE (READFILE FACESTREAM))
            (MAINW (WINDOWPROP MENU 'MAINWINDOW]
           (CLOSEF? IDSTREAM)
           (CLOSEF? FACESTREAM)
           (CLOSEF? LINESTREAM)
           (TEDIT.INSERT.OBJECT (DANCEROBJ.CREATE FORMATION FACE IDS LINES)
                  (TEXTSTREAM MAINW))
           (TTY.PROCESS (WINDOWPROP MAINW 'PROCESS])
```

(DANCEROBJ.FACEBUTTONFN

```
  [LAMBDA (ITEM MENU BUTTON)                                     (* jds "30-Dec-85 12:05")

               (* * He hit the CREATE button. Create him an object.)

    (PROG* [(MENUSTATE (FM.READSTATE MENU))
            (FACEITEM (FM.ITEMFROMID MENU 'FACE))
            (FACE (LISTGET MENUSTATE 'FACE]
           (PRINT (CONCAT FACE " " (FM.ITEMPROP ITEM 'ID))
                  PROMPTWINDOW)
           (FM.CHANGESTATE FACEITEM MENU (CONCAT FACE " " (FM.ITEMPROP ITEM 'ID])
```

(DANCEROBJ.MAKEFMDESC

```
  [LAMBDA NIL                                                    ; Edited 20-Apr-88 00:27 by

;;; Create the Free-menu description for future dancerobj menus

    (SETQ DANCEROBJ.FMDESC `(((TYPE MOMENTARY LABEL CREATE FONT (MODERN 10 BOLD)
                                     ID CREATOR SELECTEDFN DANCEROBJ.CREATEBUTTONFN)
                               (TYPE TOGGLE LABEL DONE FONT (MODERN 10 BOLD)
                                     ID FINISHER SELECTEDFN NILL)
                               (TYPE STATE LABEL "Formation: " ID FORMATION MENUITEMS %,
                                     (for FORMATION in DANCEROBJ.FORMATIONINFO by (CDDR DANCEROBJ.FORMATIONINFO)
                                        collect FORMATION)
                                     LINKS
                                     (DISPLAY FRMN))
                               (TYPE DISPLAY ID FRMN LABEL "" BOX 1 MAXWIDTH 150))
                              ((TYPE EDITSTART LABEL ID's%: LINKS (EDIT IDENT))
                               (TYPE EDIT ID IDENT LABEL ""))
                              ((TYPE EDITSTART LABEL "Facing: " LINKS (EDIT FACE))
                               (TYPE EDIT ID FACE LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux1))
                               (TYPE EDIT ID Aux1 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux2))
                               (TYPE EDIT ID Aux2 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux3))
                               (TYPE EDIT ID Aux3 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux4))
                               (TYPE EDIT ID Aux4 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux5))
                               (TYPE EDIT ID Aux5 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux6))
                               (TYPE EDIT ID Aux6 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux7))
                               (TYPE EDIT ID Aux7 LABEL ""))
                              ((TYPE EDITSTART LABEL |Aux Line:| LINKS (EDIT Aux8))
                               (TYPE EDIT ID Aux8 LABEL ""])
```

(DANCEROBJ.SORTLOCS

```
  [LAMBDA (LOCLIST)                                              (* jds "30-Dec-85 15:51")

               (* * Sort the locations of the dancers so they're numbered from bottom to top, left to right)

    (SORT LOCLIST (FUNCTION (LAMBDA (LOC1 LOC2)
                               (LET ((X1 (CAR LOC1))
                                     (Y1 (CDR LOC1))
                                     (X2 (CAR LOC2))
                                     (Y2 (CDR LOC2)))
                                    (COND
                                       ((LESSP Y1 Y2)            (* Sort first so that lower dancers come first)
                                        T)
                                       ((EQP Y1 Y2)              (* Then dancers on the same level are sorted left to right)
                                        (LESSP X1 X2])
```

(DANCEROBJ.ROT90

```
  [LAMBDA (FORMATION)                                            (* jds "31-Dec-85 10:09")

               (* * Rotate a formation by 90 degrees (doesn't preserve identifications!))

    (LIST (CADR FORMATION)
          (CAR FORMATION)
          (DANCEROBJ.SORTLOCS (for LOC in (CADDR FORMATION) collect (CONS (CDR LOC)
                                                                          (CAR LOC])
```

(DANCEROBJ.COMPOUND

```
  [LAMBDA (FORMLIST)                                          (* jds "31-Dec-85 10:09")

          (* * Given a formation info spec that is a compound of existing specs, create the fully expanded form of the formation info.)

    (COND
       ((ATOM FORMLIST)
        (LISTGET DANCEROBJ.FORMATIONINFO FORMLIST))
       ((NUMBERP (CAR FORMLIST))                              (* This is a fully-qualified formation info.
                                                              Just use it)

        FORMLIST)
       (T (PROG* ((WIDTH 0)
                  (HEIGHT 0)
                  SIZE LOCS)
                 (SELECTQ (CAR FORMLIST)
                     (BESIDE [bind SUBFORM for FORM in (CDR FORMLIST)
                                do (SETQ SUBFORM (DANCEROBJ.COMPOUND (LISTGET DANCEROBJ.FORMATIONINFO FORM)))
                                   [SETQ LOCS (APPEND LOCS (for LOC in (CADDR SUBFORM)
                                                              collect (CONS (PLUS WIDTH (CAR LOC))
                                                                            (CDR LOC]
                                   (add WIDTH (CAR SUBFORM))
                                   (SETQ HEIGHT (MAX HEIGHT (CADR SUBFORM])
                     (ONTOP [bind SUBFORM for FORM in (CDR FORMLIST)
                               do (SETQ SUBFORM (DANCEROBJ.COMPOUND (LISTGET DANCEROBJ.FORMATIONINFO FORM)))
                                  [SETQ LOCS (APPEND LOCS (for LOC in (CADDR SUBFORM)
                                                             collect (CONS (CAR LOC)
                                                                           (PLUS HEIGHT (CDR LOC]
                                  (add HEIGHT (CADR SUBFORM))
                                  (SETQ WIDTH (MAX WIDTH (CAR SUBFORM])
                     (ROTATE [RETURN (DANCEROBJ.ROT90 (DANCEROBJ.COMPOUND (CADR FORMLIST])
                     (SHOULDNT))
                 (RETURN (LIST WIDTH HEIGHT (DANCEROBJ.SORTLOCS LOCS])
)
```

;; Auxiliary Line functions

```
(DEFINEQ
```

(DANCER-Y

```
  [LAMBDA (DANCER# SPOTS TOP-BOT SIZE CURY YDESC)             ; Edited 20-Apr-88 00:36 by
    (LET [(BASE-Y (COND
                     ((LISTP DANCER#)
                      (IQUOTIENT [for D# in DANCER# sum (+ CURY YDESC (FIXR (FTIMES SIZE
                                                                                    (fetch YCOORD
                                                                                       of (CL:NTH (CL:1- D#)
                                                                                                  SPOTS]
                             (FLENGTH DANCER#)))
                     (T (+ CURY YDESC (FIXR (FTIMES SIZE (fetch YCOORD of (CL:NTH (CL:1- DANCER#)
                                                                                  SPOTS]
         (SELECTQ TOP-BOT
             (BOTTOM BASE-Y)
             (TOP (+ BASE-Y SIZE))
             (CENTER (+ BASE-Y (IQUOTIENT SIZE 2)))
             (HELP])
```

(DANCER-X

```
  [LAMBDA (DANCER# SPOTS LEFT-RT SIZE CURX)                   ; Edited 20-Apr-88 00:35 by
    ;;  Given a dancer number, compute the X location of its LEFT_RT edge.  If DANCER# is a list, compute the average such X coordinate.
    (LET [(BASE-X (COND
                     ((LISTP DANCER#)
                      (IQUOTIENT [for D# in DANCER# sum (+ CURX (FIXR (FTIMES SIZE (fetch XCOORD
                                                                                       of (CL:NTH (CL:1- D#)
                                                                                                  SPOTS]
                             (FLENGTH DANCER#)))
                     (T (+ CURX (FIXR (FTIMES SIZE (fetch XCOORD of (CL:NTH (CL:1- DANCER#)
                                                                            SPOTS]
         (SELECTQ LEFT-RT
             (LEFT BASE-X)
             (RIGHT (+ BASE-X SIZE))
             (CENTER (+ BASE-X (IQUOTIENT SIZE 2)))
             (HELP])
```

(DANCEROBJ.COMPUTE-AUXLINE-EXTENSIONS

```
  [LAMBDA (IMAGEOBJ IMAGE.STREAM)                             ; Edited 16-Apr-88 02:06 by
    ;; Computes how much the aux lines add to the 4 sides of the image object, if any.  Used to compute offsets for display, and imagebox correctiosn.
    ;; Returns a list of 4 values, representing the increments at the left bottom right and top edges, resp.
    (LET* ((FONT (FONTCREATE 'DANCER *DANCER-FONT-SIZE* NIL NIL IMAGE.STREAM))
           [SIZE (IMAX (CHARWIDTH 97 FONT)
                       (FONTPROP FONT 'HEIGHT]
           (FORMATION (IMAGEOBJPROP IMAGEOBJ 'FORMATION))
           (FINFO (LISTGET DANCEROBJ.FORMATIONINFO FORMATION))
           (WIDTH (FIXR (FTIMES (fetch (FORMATION FWIDTH) of FINFO)
                               SIZE)))
           (HEIGHT (FIXR (FTIMES (fetch (FORMATION FHEIGHT) of FINFO)
                                SIZE)))
           (DANCER-LOCATIONS (fetch (FORMATION FLOCS) of FINFO))
           (AUXILIARY-LINES (IMAGEOBJPROP IMAGEOBJ 'LINES))
           (MAX-X WIDTH)
           (MIN-X 0)
           (MAX-Y HEIGHT)
           (MIN-Y 0))
          (bind DANCER1 DANCER2 for AUX-LINE in AUXILIARY-LINES
             do ;; Run thru the auxiliary lines we're to draw, and paint them.

                (SETQ DANCER1 (fetch (DANCER-AUX-LINE DANCER1) of AUX-LINE))
                (SETQ DANCER2 (fetch (DANCER-AUX-LINE DANCER2) of AUX-LINE))
                (SETQ DASHING (fetch (DANCER-AUX-LINE DASHING) of AUX-LINE))
                (SELECTQ (fetch (DANCER-AUX-LINE LINETYPE) of AUX-LINE)
                    (TOP-SEMI [SETQ MAX-Y (IMAX MAX-Y (+ (IAVG (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE 0 0
                                                                     )
                                                               (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE 0 0
                                                                     ))
                                                         (IQUOTIENT (IABS (- (DANCER-X DANCER1 DANCER-LOCATIONS
                                                                                    'CENTER SIZE 0)
                                                                             (DANCER-X DANCER2 DANCER-LOCATIONS
                                                                                    'CENTER SIZE 0)))
                                                                3])
                    (BOT-SEMI [SETQ MIN-Y (IMIN MIN-Y (- (IAVG (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE
                                                                     0 0)
                                                               (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE
                                                                     0 0))
                                                         (IQUOTIENT (IABS (- (DANCER-X DANCER1 DANCER-LOCATIONS
                                                                                    'CENTER SIZE 0)
                                                                             (DANCER-X DANCER2 DANCER-LOCATIONS
                                                                                    'CENTER SIZE 0)))
                                                                3])
                    (L-SEMI [SETQ MIN-X (IMIN MIN-X (- (IAVG (DANCER-X DANCER1 DANCER-LOCATIONS 'LEFT SIZE 0)
                                                             (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE 0))
                                                       (IQUOTIENT (IABS (- (DANCER-Y DANCER1 DANCER-LOCATIONS
                                                                                  'CENTER SIZE 0 0)
                                                                           (DANCER-Y DANCER2 DANCER-LOCATIONS
                                                                                  'CENTER SIZE 0 0)))
                                                              3])
                    (R-SEMI [SETQ MAX-X (IMAX MAX-X (+ (IAVG (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE 0)
                                                             (DANCER-X DANCER2 DANCER-LOCATIONS 'RIGHT SIZE 0))
                                                       (IQUOTIENT (IABS (- (DANCER-Y DANCER1 DANCER-LOCATIONS
                                                                                  'CENTER SIZE 0 0)
                                                                           (DANCER-Y DANCER2 DANCER-LOCATIONS
                                                                                  'CENTER SIZE 0 0)))
                                                              3])
                    (UL-ARC)
                    (UR-ARC)
                    (LL-ARC)
                    (LR-ARC)
                    (VLINE)
                    (HLINE)
                    (DRLINE)
                    (URLINE)
                    NIL))
          (LIST (IABS MIN-X)
                (IABS MIN-Y)
                (- MAX-X WIDTH)
                (- MAX-Y HEIGHT])
```

(IAVG

```
  [LAMBDA (VAL1 VAL2)
    (IQUOTIENT (IPLUS VAL1 VAL2)
           2])

)

(DEFINEQ
```

(DANCEROBJ.X-SCALED-CURVE

```
  [LAMBDA (STREAM KNOTS X1 X2 Y)                                        ; Edited 16-Apr-88 02:14 by

    ;; Draw a curve from KNOTS that runs from X1 to X2, and is based at Y.

    (LET ((DX (IABS (- X1 X2)))
          (X0 (IMIN X1 X2)))
         (DRAWCURVE [for K in KNOTS collect (CONS [+ X0 (FIXR (FTIMES DX (CAR K]
                                                  (+ Y (FIXR (FTIMES DX (CDR K]
                NIL
                (FIXR (DSPSCALE NIL STREAM))
                NIL STREAM])
```

(DANCEROBJ.Y-SCALED-CURVE

```
  [LAMBDA (STREAM KNOTS Y1 Y2 X)                                    ; Edited 16-Apr-88 02:42 by
    (LET ((DY (IABS (- Y1 Y2)))
          (Y0 (IMIN Y1 Y2)))
         (DRAWCURVE [for K in KNOTS collect (CONS [+ X (FIXR (FTIMES DY (CAR K]
                                                  (+ Y0 (FIXR (FTIMES DY (CDR K]
                NIL
                (FIXR (DSPSCALE NIL STREAM))
                NIL STREAM])
```

(DANCEROBJ.BOTH-SCALED-CURVE

```
  [LAMBDA (STREAM KNOTS X1 Y1 X2 Y2)                                ; Edited 16-Apr-88 02:16 by
    (LET ((DX (IABS (- X2 X1)))
          (DY (IABS (- Y2 Y1)))
          (X0 (IMIN X1 X2))
          (Y0 (IMIN Y1 Y2)))
         (DRAWCURVE [for K in KNOTS collect (CONS [+ X0 (FIXR (FTIMES DX (CAR K]
                                                  (+ Y0 (FIXR (FTIMES DY (CDR K]
                NIL
                (FIXR (DSPSCALE NIL STREAM))
                NIL STREAM])
```

(DANCEROBJ.DISPLAY-AUX-LINES

```
  [LAMBDA (AUXILIARY-LINES IMAGE.STREAM DANCER-LOCATIONS SIZE CURX CURY FONTDESCENT YDESC)
                                                                    ; Edited 20-Apr-88 02:35 by

    ;; Display a dancer diagram on some image stream

    (LET* [(SCALE (DSPSCALE NIL IMAGE.STREAM))
           (LINE-BRUSH (IMAX 1 (FIXR SCALE]
          (bind DANCER1 DANCER2 DASHING for AUX-LINE in AUXILIARY-LINES
             do 
                ;; Run thru the auxiliary lines we're to draw, and paint them.

                (SETQ DANCER1 (fetch (DANCER-AUX-LINE DANCER1) of AUX-LINE))
                (SETQ DANCER2 (fetch (DANCER-AUX-LINE DANCER2) of AUX-LINE))
                [SETQ DASHING (FOR SEGMENT IN (fetch (DANCER-AUX-LINE DASHING) of AUX-LINE)
                                 COLLECT (FIXR (FTIMES SEGMENT SCALE]
                (SELECTQ (fetch (DANCER-AUX-LINE LINETYPE) of AUX-LINE)
                     (TOP-SEMI (DANCEROBJ.X-SCALED-CURVE IMAGE.STREAM DANCEROBJ.TOP-KNOTS (DANCER-X
                                                                                           DANCER1
                                                                                           DANCER-LOCATIONS
                                                                                           'CENTER SIZE CURX)
                                      (DANCER-X DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURX)
                                      (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)))
                     (ITOP-SEMI (DANCEROBJ.X-SCALED-CURVE IMAGE.STREAM DANCEROBJ.TOP-KNOTS
                                       (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                          FONTDESCENT)
                                       (+ FONTDESCENT (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX))
                                       (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)))
                     ((BOTTOM-SEMI BOT-SEMI)
                          (DANCEROBJ.X-SCALED-CURVE IMAGE.STREAM DANCEROBJ.BOT-KNOTS (DANCER-X DANCER1
                                                                                      DANCER-LOCATIONS
                                                                                      'CENTER SIZE CURX)
                                 (DANCER-X DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURX)
                                 (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)))
                     ((IBOTTOM-SEMI IBOT-SEMI)
                          (DANCEROBJ.X-SCALED-CURVE IMAGE.STREAM DANCEROBJ.BOT-KNOTS (- (DANCER-X DANCER1
                                                                                         DANCER-LOCATIONS
                                                                                         'RIGHT SIZE CURX
                                                                                         )
                                                                                      FONTDESCENT)
                                 (+ FONTDESCENT (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX))
                                 (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)))
                     ((LEFT-SEMI L-SEMI)
                          (DANCEROBJ.Y-SCALED-CURVE IMAGE.STREAM DANCEROBJ.L-KNOTS (DANCER-Y DANCER1
                                                                                    DANCER-LOCATIONS
                                                                                    'CENTER SIZE CURY
                                                                                    YDESC)
                                 (DANCER-Y DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURY YDESC)
                                 (DANCER-X DANCER1 DANCER-LOCATIONS 'LEFT SIZE CURX)))
                     (IL-SEMI (DANCEROBJ.Y-SCALED-CURVE IMAGE.STREAM DANCEROBJ.L-KNOTS
                                     (- (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)
                                     (+ FONTDESCENT (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC))
                                     (+ FONTDESCENT (DANCER-X DANCER1 DANCER-LOCATIONS 'LEFT SIZE CURX))))
                     ((RIGHT-SEMI R-SEMI)
                          (DANCEROBJ.Y-SCALED-CURVE IMAGE.STREAM DANCEROBJ.R-KNOTS (DANCER-Y DANCER1
                                                                                    DANCER-LOCATIONS
                                                                                    'CENTER SIZE CURY
                                                                                    YDESC)
                                 (DANCER-Y DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURY YDESC)
                                 (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)))
                     (IR-SEMI (DANCEROBJ.Y-SCALED-CURVE IMAGE.STREAM DANCEROBJ.R-KNOTS
                                     (- (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)
                                     (+ FONTDESCENT (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC))
                                     (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                        FONTDESCENT)))
                     (UL-ARC ;; ARCS ARE ALWAYS SPECIFIED CLOCKWISE!
                             (DANCEROBJ.BOTH-SCALED-POLY IMAGE.STREAM DANCEROBJ.LU-KNOTS
                                     (+ FONTDESCENT (DANCER-X DANCER1 DANCER-LOCATIONS 'LEFT SIZE CURX))
                                     (- (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)
                                     (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX)
                                     (- (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)))
                     (UR-ARC (DANCEROBJ.BOTH-SCALED-CURVE IMAGE.STREAM DANCEROBJ.RU-KNOTS
                                     (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                        FONTDESCENT)
                                     (- (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)
                                     (- (DANCER-X DANCER2 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                        FONTDESCENT)
                                     (- (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)))
                     (LR-ARC (DANCEROBJ.BOTH-SCALED-CURVE IMAGE.STREAM DANCEROBJ.RB-KNOTS
                                     (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                        FONTDESCENT)
                                     (+ (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                        FONTDESCENT)
                                     (- (DANCER-X DANCER2 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                        FONTDESCENT)
                                     (+ (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                        FONTDESCENT)))
                     (LL-ARC (DANCEROBJ.BOTH-SCALED-CURVE IMAGE.STREAM DANCEROBJ.LB-KNOTS
                                     (+ (DANCER-X DANCER1 DANCER-LOCATIONS 'LEFT SIZE CURX)
                                        FONTDESCENT)
                                     (+ (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                        FONTDESCENT)
                                     (+ (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX)
                                        FONTDESCENT)
                                     (+ (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                        FONTDESCENT)))
                     (VLINE ;; Draw a line DOWN from dancer 1 to 2.
                            (DRAWLINE (DANCER-X DANCER1 DANCER-LOCATIONS 'CENTER SIZE CURX)
                                   (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                   (DANCER-X DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURX)
                                   (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                   LINE-BRUSH
                                   'PAINT IMAGE.STREAM NIL DASHING))
                     (IVLINE ;; Draw a line DOWN from dancer 1 to 2--including the area inside the nose border.
                             (DRAWLINE (DANCER-X DANCER1 DANCER-LOCATIONS 'CENTER SIZE CURX)
                                     (+ FONTDESCENT (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC))
                                     (DANCER-X DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURX)
                                     (- (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                        FONTDESCENT)
                                     LINE-BRUSH
                                     'PAINT IMAGE.STREAM NIL DASHING))
                     (HLINE ;; Draw a line RIGHT from dancer 1 to 2
                            (DRAWLINE (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                   (DANCER-Y DANCER1 DANCER-LOCATIONS 'CENTER SIZE CURY YDESC)
                                   (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX)
                                   (DANCER-Y DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURY YDESC)
                                   LINE-BRUSH
                                   'PAINT IMAGE.STREAM NIL DASHING))
                     (IHLINE ;; Draw a line RIGHT from dancer 1 to 2
                             (DRAWLINE (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                          FONTDESCENT)
                                     (DANCER-Y DANCER1 DANCER-LOCATIONS 'CENTER SIZE CURY YDESC)
                                     (+ FONTDESCENT (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX))
                                     (DANCER-Y DANCER2 DANCER-LOCATIONS 'CENTER SIZE CURY YDESC)
                                     LINE-BRUSH
                                     'PAINT IMAGE.STREAM NIL DASHING))
                     (DRLINE ;; Draw a line DOWN & RIGHT from dancer 1 to 2.
                             (DRAWLINE (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                     (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                     (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX)
                                     (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                     LINE-BRUSH
                                     'PAINT IMAGE.STREAM NIL DASHING))
                     (IDRLINE ;; Draw a line DOWN & RIGHT from dancer 1 to 2. to the edge of the pix.
                              (DRAWLINE (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                           FONTDESCENT)
```

```
                                            (+ FONTDESCENT (DANCER-Y DANCER1 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC))
                                            (+ FONTDESCENT (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX))
                                            (- (DANCER-Y DANCER2 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                               FONTDESCENT)
                                            LINE-BRUSH
                                            'PAINT IMAGE.STREAM NIL DASHING))
                            (URLINE ;; Draw a line UP & RIGHT from dancer 1 to 2.

                                    (DRAWLINE (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                           (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                           (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX)
                                           (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC)
                                           LINE-BRUSH
                                           'PAINT IMAGE.STREAM NIL DASHING))
                            (IURLINE ;; Draw a line UP & RIGHT from dancer 1 to 2.

                                     (DRAWLINE (- (DANCER-X DANCER1 DANCER-LOCATIONS 'RIGHT SIZE CURX)
                                                  FONTDESCENT)
                                            (- (DANCER-Y DANCER1 DANCER-LOCATIONS 'TOP SIZE CURY YDESC)
                                               FONTDESCENT)
                                            (+ FONTDESCENT (DANCER-X DANCER2 DANCER-LOCATIONS 'LEFT SIZE CURX))
                                            (+ FONTDESCENT (DANCER-Y DANCER2 DANCER-LOCATIONS 'BOTTOM SIZE CURY YDESC))
                                            LINE-BRUSH
                                            'PAINT IMAGE.STREAM NIL DASHING))
                            (HELP])

)

(DEFINEQ

(DANCEROBJ.X-SCALED-POLY
  [LAMBDA (STREAM KNOTS X1 X2 Y0 DASHING)                    ; Edited 16-Apr-88 12:32 by

    ;; Draw a curve from KNOTS that runs from X1 to X2, and is based at Y.

    (LET [(DX (IABS (- X1 X2)))
          (X0 (IMIN X1 X2))
          (SCALED-DASHING (FOR LEN IN DASHING COLLECT (FIXR (FTIMES LEN (DSPSCALE NIL STREAM]
         (bind [X _ (+ X0 (FIXR (FTIMES DX (CAAR KNOTS]
               [Y _ (+ Y0 (FIXR (FTIMES DY (CDAR KNOTS] for K in (CDR KNOTS)
            do (DRAWLINE X Y [SETQ X (+ X0 (FIXR (FTIMES DX (CAR K]
                      [SETQ Y (+ Y0 (FIXR (FTIMES DY (CDR K]
                      (FIXR (DSPSCALE NIL STREAM))
                      'PAINT STREAM NIL SCALED-DASHING))
         (DRAWCURVE [for K in KNOTS collect (CONS [+ X0 (FIXR (FTIMES DX (CAR K]
                                                  (+ Y (FIXR (FTIMES DX (CDR K]
                NIL
                (FIXR (DSPSCALE NIL STREAM))
                NIL STREAM])

(DANCEROBJ.BOTH-SCALED-POLY
  [LAMBDA (STREAM KNOTS X1 Y1 X2 Y2 DASHING)                 ; Edited 16-Apr-88 12:27 by
    (LET [(DX (IABS (- X1 X2)))
          (DY (IABS (- Y1 Y2)))
          (X0 (IMIN X1 X2))
          (Y0 (IMIN Y1 Y2))
          (SCALED-DASHING (for LEN in DASHING collect (FIXR (FTIMES LEN (DSPSCALE NIL STREAM]
         (bind [X _ (+ X0 (FIXR (FTIMES DX (CAAR KNOTS]
               [Y _ (+ Y0 (FIXR (FTIMES DY (CDAR KNOTS] for K in (CDR KNOTS)
            do (DRAWLINE X Y [SETQ X (+ X0 (FIXR (FTIMES DX (CAR K]
                      [SETQ Y (+ Y0 (FIXR (FTIMES DY (CDR K]
                      (FIXR (DSPSCALE NIL STREAM))
                      'PAINT STREAM NIL SCALED-DASHING])

)

;; Knots for semi-circles (actually 30% ellipses)

(RPAQQ DANCEROBJ.TOP-KNOTS ((8.940697E-8 . 2.1457673E-7)
                            (0.030153751 . 0.10260624)
                            (0.11697778 . 0.1928364)
                            (0.24999994 . 0.25980768)
                            (0.41317588 . 0.29544234)
                            (0.586824 . 0.2954423)
                            (0.74999994 . 0.2598076)
                            (0.8830222 . 0.19283625)
                            (0.9698462 . 0.10260604)
                            (0.99999994 . 0.0)))

(RPAQQ DANCEROBJ.BOT-KNOTS ((8.940697E-8 . -2.1457673E-7)
                            (0.030153751 . -0.10260624)
                            (0.11697778 . -0.1928364)
                            (0.24999994 . -0.25980768)
                            (0.41317588 . -0.29544234)
                            (0.586824 . -0.2954423)
                            (0.74999994 . -0.2598076)
                            (0.8830222 . -0.19283625)
```

```
                                 (0.9698462 . -0.10260604)
                                 (0.99999994 . 0.0)))

(RPAQQ DANCEROBJ.L-KNOTS ((-2.1457673E-7 . 8.940697E-8)
                          (-0.10260624 . 0.030153751)
                          (-0.1928364 . 0.11697778)
                          (-0.25980768 . 0.24999994)
                          (-0.29544234 . 0.41317588)
                          (-0.2954423 . 0.586824)
                          (-0.2598076 . 0.74999994)
                          (-0.19283625 . 0.8830222)
                          (-0.10260604 . 0.9698462)
                          (0.0 . 0.99999994)))

(RPAQQ DANCEROBJ.R-KNOTS ((2.1457673E-7 . 8.940697E-8)
                          (0.10260624 . 0.030153751)
                          (0.1928364 . 0.11697778)
                          (0.25980768 . 0.24999994)
                          (0.29544234 . 0.41317588)
                          (0.2954423 . 0.586824)
                          (0.2598076 . 0.74999994)
                          (0.19283625 . 0.8830222)
                          (0.10260604 . 0.9698462)
                          (0.0 . 0.99999994)))

;; 1/4 circle & ellipse quadrant knots

(RPAQQ DANCEROBJ.LU-KNOTS ((1.7881393E-7 . -1.1920929E-7)
                           (0.01519233 . 0.17364804)
                           (0.060307562 . 0.34201998)
                           (0.13397467 . 0.49999985)
                           (0.23395562 . 0.6427874)
                           (0.35721248 . 0.7660443)
                           (0.5 . 0.86602527)
                           (0.65797985 . 0.9396924)
                           (0.8263518 . 0.98480755)
                           (1.0 . 0.9999999)))

(RPAQQ DANCEROBJ.LB-KNOTS ((1.7881393E-7 . 1.0000001)
                           (0.01519233 . 0.82635194)
                           (0.060307562 . 0.65798)
                           (0.13397467 . 0.5000001)
                           (0.23395562 . 0.3572126)
                           (0.35721248 . 0.23395568)
                           (0.5 . 0.13397473)
                           (0.65797985 . 0.060307622)
                           (0.8263518 . 0.015192449)
                           (1.0 . 1.1920929E-7)))

(RPAQQ DANCEROBJ.RU-KNOTS ((0.0 . 0.9999999)
                           (0.17364818 . 0.98480755)
                           (0.34202012 . 0.9396924)
                           (0.49999997 . 0.86602527)
                           (0.6427875 . 0.7660443)
                           (0.7660444 . 0.6427874)
                           (0.8660253 . 0.49999985)
                           (0.93969244 . 0.34201998)
                           (0.98480767 . 0.17364804)
                           (0.9999998 . -1.1920929E-7)))

(RPAQQ DANCEROBJ.RB-KNOTS ((0.0 . 1.1920929E-7)
                           (0.17364818 . 0.015192449)
                           (0.34202012 . 0.060307622)
                           (0.49999997 . 0.13397473)
                           (0.6427875 . 0.23395568)
                           (0.7660444 . 0.3572126)
                           (0.8660253 . 0.5000001)
                           (0.93969244 . 0.65798)
                           (0.98480767 . 0.82635194)
                           (0.9999998 . 1.0000001)))

(DECLARE%: EVAL@COMPILE

(RECORD DANCER-AUX-LINE (LINETYPE DANCER1 DANCER2 DASHING))
)

(RPAQQ *DANCER-FONT-SIZE* 12)

(RPAQQ DANCEROBJ.MENU NIL)

(RPAQQ DANCEROBJ.FORMATIONINFO NIL)

(RPAQQ DANCEROBJ.FMDESC NIL)

(RPAQ DANCEROBJ.TERMTABLE (DANCEROBJ.MAKETERMTABLE))

(RPAQQ DANCEROBJ.INITIAL.FORMATION.INFO
       (DIAMOND (2 4 ((0.5 . 0)
                      (0 . 1.5)
                      (1 . 1.5)
                      (0.5 . 3)))
              DIAMONDR90
              (ROTATE DIAMOND)
              DIAMONDS
              (BESIDE DIAMOND DIAMOND)
              DIAMONDSR90
              (ROTATE DIAMONDS)
              PPDIAMONDS
              (BESIDE DIAMONDR90 DIAMONDR90)
              PPDIAMONDSR90
              (ROTATE PPDIAMONDS)
              O/R% Diamonds
              (6 2 ((0 . 0.5)
                    (1 . 0.5)
                    (2 . 0)
                    (3 . 0)
                    (2 . 1)
                    (3 . 1)
                    (4 . 0.5)
                    (5 . 0.5)))
              |O/R Diamonds R90|
              (ROTATE O/R% Diamonds)
              T/Diamonds
              (BESIDE DIAMOND DIAMOND DIAMOND)
              |T/Diamonds R 90|
              (ROTATE T/Diamonds)
              DTHAR
              (4 4 ((0.5 . 0.5)
                    (2.5 . 0.5)
                    (1.1 . 1.1)
                    (1.9 . 1.1)
                    (1.1 . 1.9)
                    (1.9 . 1.9)
                    (0.5 . 2.5)
                    (2.5 . 2.5)))
              RTHAR
              (4 4 ((1.5 . 0)
                    (1.5 . 1)
                    (0 . 1.5)
                    (1 . 1.5)
                    (2 . 1.5)
                    (3 . 1.5)
                    (1.5 . 2)
                    (1.5 . 3)))
              THAR
              (5 5 ((2 . 0)
                    (2 . 1)
                    (0 . 2)
                    (1 . 2)
                    (3 . 2)
                    (4 . 2)
                    (2 . 3)
                    (2 . 4)))
              HRGLASS
              (4 4 ((1.5 . 0)
                    (0 . 0.75)
                    (3 . 0.75)
                    (1 . 1.5)
                    (2 . 1.5)
                    (0 . 2.25)
                    (3 . 2.25)
                    (1.5 . 3)))
              GALAXY
              (4 4 ((1.5 . 0)
                    (1 . 1)
                    (2 . 1)
                    (0 . 1.5)
                    (3 . 1.5)
                    (1 . 2)
                    (2 . 2)
                    (1.5 . 3)))
              1X4
              (4 1 ((0 . 0)
                    (1 . 0)
                    (2 . 0)
                    (3 . 0)))
              4X1
              (ROTATE 1X4)
              2X2
              (2 2 ((0 . 0)
                    (1 . 0)
                    (0 . 1)
                    (1 . 1)))
              2X4
              (4 2 ((0 . 0)
                    (1 . 0)
                    (2 . 0)
                    (3 . 0)
                    (0 . 1)
                    (1 . 1)
                    (2 . 1)
                    (3 . 1)))
              4X2
              (ROTATE 2X4)
              1X8
              (BESIDE 1X4 1X4)
              8X1
              (ROTATE 1X8)
              1X3
              (3 1 ((0 . 0)
                    (1 . 0)
                    (2 . 0)))
              3X1
              (ROTATE 1X3)
              3X2
              (BESIDE 3X1 3X1)
              2X3
              (ROTATE 3X2)
              3X4
              (BESIDE 3X2 3X2)
              4X3
              (ROTATE 3X4)
              1X2
              (2 1 ((0 . 0)
                    (1 . 0)))
              2X1
              (ROTATE 1X2)
              2X6
              (BESIDE 2X2 2X2 2X2)
              6X2
              (ROTATE 2X6)
              2X5
              (BESIDE 2X2 2X2 2X1)
              5X2
              (ROTATE 2X5)
              1/4TAG
              (4 3 ((1 . 0)
                    (2 . 0)
                    (0 . 1)
                    (1 . 1)
                    (2 . 1)
                    (3 . 1)
                    (1 . 2)
                    (2 . 2)))
              |1/4TAG R 90|
              (ROTATE 1/4TAG)
              O
              (4 4 ((1 . 0)
                    (2 . 0)
                    (0 . 1)
                    (3 . 1)
                    (0 . 2)
                    (3 . 2)
                    (1 . 3)
                    (2 . 3)))
              BFLY
              (4 4 ((0 . 0)
                    (3 . 0)
                    (1 . 1)
                    (2 . 1)
                    (1 . 2)
                    (2 . 2)
                    (0 . 3)
                    (3 . 3)))
              H
              (4 3 ((0 . 0)
                    (3 . 0)
                    (0 . 1)
                    (1 . 1)
                    (2 . 1)
                    (3 . 1)
                    (0 . 2)
                    (3 . 2)))
              H% R90
              (ROTATE H)
              LBLOX
              (4 4 ((1 . 0)
                    (3 . 0)
                    (0 . 1)
                    (2 . 1)
                    (1 . 2)
```

```
                (3 . 2)
                (0 . 3)
                (2 . 3)))
          RBLOX
          (4 4 ((0 . 0)
                (2 . 0)
                (1 . 1)
                (3 . 1)
                (0 . 2)
                (2 . 2)
                (1 . 3)
                (3 . 3)))
          SET
          (4 4 ((1 . 0)
                (2 . 0)
                (0 . 1)
                (3 . 1)
                (0 . 2)
                (3 . 2)
                (1 . 3)
                (2 . 3)))
          PHANT
          (4 4 ((0 . 0.5)
                (1 . 0.5)
                (2.5 . 0)
                (2.5 . 1)
                (0.5 . 2)
                (0.5 . 3)
                (2 . 2.5)
                (3 . 2.5)))
          VHPHANT
          (ROTATE PHANT)
          |Wv bet Vmw|
          (4 6 ((1.5 . 0)
                (1.5 . 1)
                (0 . 2.5)
                (1 . 2.5)
                (2 . 2.5)
                (3 . 2.5)
                (1.5 . 4)
                (1.5 . 5)))
          |Wv bet Vmw R 90|
          (ROTATE |Wv bet Vmw|)
          |Vdi bet Vmw|
          (2 7 ((0.5 . 0)
                (0.5 . 1)
                (0.5 . 2)
                (0 . 3)
                (1 . 3)
                (0.5 . 4)
                (0.5 . 5)
                (0.5 . 6)))
          1-3-3-1
          (3 4 ((1 . 0)
                (0 . 1)
                (1 . 1)
                (2 . 1)
                (0 . 2)
                (1 . 2)
                (2 . 2)
                (1 . 3)))
          1-3-3-1% R90
          (ROTATE 1-3-3-1)
          |Ac Duc 1 1/2|
          (4 6 ((1.5 . 0)
                (1.5 . 1)
                (1.5 . 2)
                (0 . 2.5)
                (3 . 2.5)
                (1.5 . 3)
                (1.5 . 4)
                (1.5 . 5)))
          1-2-2-2-1
          (2 5 ((0.5 . 0)
                (0 . 1)
                (1 . 1)
                (0 . 2)
                (1 . 2)
                (0 . 3)
                (1 . 3)
                (0.5 . 4)))
          1-2-2-2-1% horiz
          (ROTATE 1-2-2-2-1)
          Star
          (2 2 ((0.5 . 0)
                (0 . 0.5)
                (1 . 0.5)
```

```
                    (0.5 . 1)))
              |Single 1/4 Tag|
              (2 3 ((0.5 . 0)
                    (0 . 1)
                    (1 . 1)
                    (0.5 . 2)))
              |Single 1/4 Tag R 90|
              (ROTATE |Single 1/4 Tag|)
              Triangle
              (2 2 ((0 . 0)
                    (1 . 0)
                    (0.5 . 1)))
              |Triangle R 90|
              (ROTATE Triangle)
              Wide% Tri
              (3 2 ((0 . 0)
                    (2 . 0)
                    (1 . 1)))
              |Wide Tri R 90|
              (ROTATE Wide% Tri)
              |L Exch 1/2|
              (5 4 ((2 . 0)
                    (0 . 0.5)
                    (1 . 0.5)
                    (2 . 1)
                    (2 . 2)
                    (3 . 2.5)
                    (4 . 2.5)
                    (2 . 3)))
              |L Exch 1/2 R 90|
              (ROTATE |L Exch 1/2|)
              |R Exch 1/2|
              (5 4 ((2 . 0)
                    (3 . 0.5)
                    (4 . 0.5)
                    (2 . 1)
                    (2 . 2)
                    (0 . 2.5)
                    (1 . 2.5)
                    (2 . 3)))
              |R Exch 1/2 R 90|
              (ROTATE |R Exch 1/2|)))

(ADDTOVAR IMAGEOBJGETFNS (DANCEROBJ.GETFN))

(ADDTOVAR IMAGEOBJGETFNS (DANCEROBJ.GETFN2))

(DECLARE%: DONTEVAL@LOAD DOCOPY

(DANCEROBJ.INIT)
)

(DECLARE%: EVAL@COMPILE

(RECORD FORMATION (FWIDTH FHEIGHT FLOCS))
)

(PUTPROPS DANCEROBJ COPYRIGHT ("Xerox Corporation" 1985 1986 1987 1988 1900))
```

{MEDLEY}<test>gc>Hand>DANCEROBJ.;1   28-Jun-2024 18:34:36

-- Listed on 30-Jun-2024 13:59:45 --

FUNCTION INDEX

```
DANCER-X ...........................................7     DANCEROBJ.GETFN ....................................3
DANCER-Y ...........................................7     DANCEROBJ.GETFN2 ...................................3
DANCERMENU.CLOSEFN .................................1     DANCEROBJ.IMAGEBOXFN ...............................3
DANCEROBJ.BOTH-SCALED-CURVE ........................9     DANCEROBJ.INIT .....................................4
DANCEROBJ.BOTH-SCALED-POLY ........................11     DANCEROBJ.INITFORMATIONINFO ........................4
DANCEROBJ.BUTTONEVENTINFN ..........................3     DANCEROBJ.MAKEFMDESC ...............................6
DANCEROBJ.COMPOUND .................................7     DANCEROBJ.MAKEMENU .................................4
DANCEROBJ.COMPUTE-AUXLINE-EXTENSIONS ...............7     DANCEROBJ.MAKETERMTABLE ............................5
DANCEROBJ.COPYFN ...................................4     DANCEROBJ.PUTFN ....................................3
DANCEROBJ.CREATE ...................................1     DANCEROBJ.ROT90 ....................................6
DANCEROBJ.CREATEBUTTONFN ...........................5     DANCEROBJ.SORTLOCS .................................6
DANCEROBJ.DISPLAY-AUX-LINES ........................9     DANCEROBJ.X-SCALED-CURVE ...........................8
DANCEROBJ.DISPLAYFN ................................2     DANCEROBJ.X-SCALED-POLY ...........................11
DANCEROBJ.EDIT .....................................4     DANCEROBJ.Y-SCALED-CURVE ...........................9
DANCEROBJ.FACEBUTTONFN .............................6     IAVG ...............................................8
```

VARIABLE INDEX

```
*DANCER-FONT-SIZE* ..............12    DANCEROBJ.L-KNOTS ...............12    DANCEROBJ.RB-KNOTS ..............12
DANCEROBJ.BOT-KNOTS .............11    DANCEROBJ.LB-KNOTS ..............12    DANCEROBJ.RU-KNOTS ..............12
DANCEROBJ.FMDESC ................12    DANCEROBJ.LU-KNOTS ..............12    DANCEROBJ.TERMTABLE .............12
DANCEROBJ.FORMATIONINFO .........12    DANCEROBJ.MENU ..................12    DANCEROBJ.TOP-KNOTS .............11
DANCEROBJ.INITIAL.FORMATION.INFO 12    DANCEROBJ.R-KNOTS ...............12    IMAGEOBJGETFNS ..................16
```

RECORD INDEX

```
DANCER-AUX-LINE .................12    FORMATION .......................16
```