

File created: 19-Jan-93 10:27:05 {DSK}<python>lde>lispcore>sources>DSPRINTDEF.;2

changes to: (RECORDS DEDITMAP)

previous date: 16-May-90 16:17:21 {DSK}<python>lde>lispcore>sources>DSPRINTDEF.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1990, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **DSPRINTDEFCOMS**

```
[ (COMS                                     ; NEWPRINTDEF primitives for a display that maintains a map
                                     ; as it PPs
  (DECLARE%: EVAL@COMPILE DONTCOPY (MACROS BLANKS WIDTH XPOSITION YPOSITION OVERLAP)
    (GLOBALVARS \DEDITDPS \DSPRINTBP \DEDITMEHASH \DEDITDPHASH \DEDITFONT# \DEDITFONTS
      COMMENTFLG)
    (CONSTANTS DOTSTRING)
    (COMS ;; The DEDITMAP record declaration is here because several functions in this file use it when MAKEMAP is true
          ;; (only on calls from DEDIT). However, it is DONTCOPY -- the INITRECORDS definition is on DEDITPP, which is
          ;; only loaded when DEDIT is
      (RECORDS DEDITMAP)))
  (FNS PRINOPEN PRINSHUT PRIN1S PRIN2S PRINENDLINE PRINDOTP SETFONT MAKEDOTPTAIL))
(COMS                                     ; Wrappers
  (FNS SUPERPRINT/WRAPPER))
(COMS                                     ; String printer
  (FNS PRIN2STRING PRIN2-LONG-STRING)
  (INITVARS (*DIVIDE-LONG-STRINGS* 'DISPLAY]))
```

:: NEWPRINTDEF primitives for a display that maintains a map as it PPs

```
(DECLARE%: EVAL@COMPILE DONTCOPY
(DECLARE%: EVAL@COMPILE
(PUTPROPS BLANKS MACRO ((N)
  (TIMES N SPACEWIDTH)))
(PUTPROPS WIDTH MACRO ((STR STREAM P2FLG)
  (STRINGWIDTH STR (OR STREAM *STANDARD-OUTPUT*)
    P2FLG)))
(PUTPROPS XPOSITION MACRO ((X)
  (DSPXPOSITION X FILE)))
(PUTPROPS YPOSITION MACRO ((Y)
  (DSPYPOSITION Y)))
(PUTPROPS OVERLAP MACRO [OPENLAMBDA (H1 L1 H2 L2)
  (NOT (OR (ILESSP H1 L2)
    (ILESSP H2 L1)))
])
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \DEDITDPS \DSPRINTBP \DEDITMEHASH \DEDITDPHASH \DEDITFONT# \DEDITFONTS COMMENTFLG)
)
(DECLARE%: EVAL@COMPILE
(RPAQQ DOTSTRING " . ")
(CONSTANTS DOTSTRING)
)
```

:: The DEDITMAP record declaration is here because several functions in this file use it when MAKEMAP is true (only on calls from DEDIT). However, it is DONTCOPY -- the INITRECORDS definition is on DEDITPP, which is only loaded when DEDIT is

```
(DECLARE%: EVAL@COMPILE
[DATATYPE DEDITMAP ((D# BYTE)
  (TAIL POINTER)
  (F# BYTE)
  (BP POINTER)
  (STARTX WORD)
  (STOPX WORD)
  (STARTY WORD)
  (STOPY WORD)
  (LONGSTRINGP FLAG)
  (LONGSTRING1MARGINP FLAG)
  ; SELEXP is a string neatly divided over several lines
```

:: String's left margin is same on every line. If false, then left margin for second and subsequent lines is STARTX of BP
(LONGSTRINGSYMMETRICP FLAG)
:: String was in a centered comment, so its right margin is indented symmetrically with its left margin

```
(NIL 5 FLAG)
(WRAPPER POINTER))
(ACCESSFNS DEDITMAP ((FNT (ELT \DEDITFONTS (fetch F# of DATUM)))
(PDSP (ELT \DEDITDSPS (fetch D# of DATUM)))
(SELEXP (CAR (fetch TAIL of DATUM)))
(LPEND (DEDIT.LPEND DATUM))
(RPSTART (DEDIT.RPSTART DATUM))
)
(/DECLAREDATATYPE 'DEDITMAP
' (BYTE POINTER BYTE POINTER WORD WORD WORD WORD FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER)
;; ---field descriptor list elided by lister---
' 12)
)
```

(DEFINEQ

(PRINOPEN

```
[LAMBDA (TAIL PAREN FILE) ; (* Imm "30-Jul-85 03:12")
[COND
(MAKEMAP (SETQ MAKEMAP (MAKEMAPENTRY TAIL (AND (NEQ MAKEMAP T)
MAKEMAP)
(DSPXPOSITION NIL FILE)
(DSPYPOSITION NIL FILE)
0 0 (\DEDITFONT# FILE]
(PRIN3 PAREN FILE])
```

(PRINSHUT

```
[LAMBDA (TAIL PAREN FILE) ; (* AJB "22-Jan-86 16:30")
(AND PAREN (PRIN3 PAREN FILE))
(COND
(MAKEMAP (COND
((EQ MAKEMAP T)
(SHOULDNT)))
(replace STOPX of MAKEMAP with (DSPXPOSITION NIL FILE))
(replace STOPY of MAKEMAP with (DSPYPOSITION NIL FILE))
(SETQ MAKEMAP (OR (fetch BP of MAKEMAP)
T]))
```

(PRIN1S

```
[LAMBDA (STR TAIL FILE) ; (* Imm "18-Jan-86 01:09")
(COND
(MAKEMAP
(MAKEMAPENTRY TAIL (AND (NEQ MAKEMAP T)
MAKEMAP)
(DSPXPOSITION NIL FILE)
(DSPYPOSITION NIL FILE)
(PROGN (PRIN3 STR FILE)
(DSPXPOSITION NIL FILE))
(DSPYPOSITION NIL FILE)
(\DEDITFONT# FILE))
STR)
(T (PRIN3 STR FILE])) ; if remembering where things went
```

(PRIN2S

```
[LAMBDA (STR TAIL FILE) ; (* Imm "30-Jul-85 03:26")
(COND
(MAKEMAP (MAKEMAPENTRY TAIL (AND (NEQ MAKEMAP T)
MAKEMAP)
(DSPXPOSITION NIL FILE)
(DSPYPOSITION NIL FILE)
(PROGN (PRIN4 STR FILE)
(DSPXPOSITION NIL FILE))
(DSPYPOSITION NIL FILE)
(\DEDITFONT# FILE))
STR)
(T (PRIN4 STR FILE]))
```

(PRINENDLINE

```
[LAMBDA (NEWXPOSITION FILE) ; Edited 1-Apr-88 14:24 by bvm
;; Terminate line, setting x at NEWXPOSITION.
(OR FILE (SETQ FILE *STANDARD-OUTPUT*))
(COND
(MAKEMAP ; From DEdit
(MOVETO NEWXPOSITION (+ (DSPYPOSITION NIL FILE)
(DSPLINEFEED NIL FILE))
```

```

      FILE))
(T (TERPRI FILE)
  (COND
    ((OR (SELECTQ (IMAGESTREAMTYPE FILE)
                  ((NIL TEXT)
                   T)
                  (PROGN
                   NIL))
         (if (EQ FILE (TTYDISPLAYSTREAM))
              then
              (DRIBBLEFILE)))
      (SETFONT (PROG1 (SETFONT DEFAULTFONT FILE)
                    ;; Print introductory spaces in the default font because we don't quite have this right yet for pspool files
                    (LET ((NS (QUOTIENT (- NEWXPOSITION (DSPXPOSITION NIL FILE)
                                                    SPACEWIDTH)))
                          (RPTQ (QUOTIENT NS 8)
                                (PRIN3 " " FILE))
                          (RPTQ (REMAINDER NS 8)
                                (PRIN3 " " FILE))))
                  FILE)))
(DSPXPOSITION NEWXPOSITION FILE])

```

; These don't know how to set x position
 ; Assume all other image streams are ok
 ; Even if FILE knows how to set xpos, the dribble file doesn't, so
 ; use spaces

(PRINDOTP

```

[LAMBDA (E FILE)
  ;; Print a dotted tail consisting of the non-list E, i.e., print ". <E>"
  (LET* [(DOT " . ")
         (MAXPOS (- RMARGIN (WIDTH E FILE T)
                       (WIDTH DOT FILE)
                       (WIDTH " " FILE)
                     (if (AND (> (DSPXPOSITION NIL FILE)
                                MAXPOS)
                          (>= MAXPOS FIRSTPOS))
                        then
                        (PRINENDLINE MAXPOS FILE))
                    (PRIN3 DOT FILE)
                    (PRIN2S E (COND
                              (MAKEMAP (MAKEDOTPTAIL E MAKEMAP))
                              (T (CONS E E)))
                      FILE])

```

; Edited 13-Apr-88 15:08 by bvm
 ; MAXPOS is the rightmost position at which this will fit
 ; Print dotted tail on next line as far to right as possible

(SETFONT

```

[LAMBDA (FONT FILE)
  ;; FONT can be a font, a number or a FONTCLASS. Returns a FONTDESCRIPTOR FOR PPDSP
  (COND
    (FONT
     (LET ((OLDFONT (DSPFONT FONT FILE)))
         [COND
          ((NEQ OLDFONT FONT)
           (AND MAKEMAP (SETQ \DEDITFONT#))
           (SETQ SPACEWIDTH (CHARWIDTH (CHARCODE x)
                                         FILE))
           )
          ((NULL SPACEWIDTH)
           ;; initialize SPACEWIDTH
           (SETQ SPACEWIDTH (CHARWIDTH (CHARCODE x)
                                         FILE))
          OLDFONT])

```

; Edited 2-Nov-88 13:16 by drc:
 ; if FONT is NIL, leave things alone.
 ; SPACEWIDTH really means "average character width"

(MAKEDOTPTAIL

```

[LAMBDA (V B)
  (* bas%: "18-Mar-84 21:10")
  ;; DPs have map entries keyed off a dummy CONS which is found by a hash link off the parent CONS thru the \DEDITDPHASH array. Done this
  ;; way so we have a CONS to push on the selection stack which makes most of the changing functions transparent and which can be found quickly
  ;; and repeatably. Usually we do not have the DP cons itself in hand, hence use of the parent CONS.
  ;; If there is a dummy CONS for the DP, we must preserve it because it may be being used as a key e.g. from the selection stack. But we must
  ;; also ensure that it has the right contents, namely V
  (PUTHASH (fetch TAIL of B)
            (RPLNODE (OR (GETHASH (fetch TAIL of B)
                                  \DEDITDPHASH)
                          (CONS))
              V V)
            \DEDITDPHASH])
)

```

)

;; Wrappers


```

(if USE-SEMI-COLONS
  then (if (< USE-SEMI-COLONS 5)
    then
      ; Semicolon comment
      [SETQ SEMIWIDTH (+ SPACEWIDTH (TIMES USE-SEMI-COLONS (CHARWIDTH (CHARCODE ";")
                                                                FONT])
      (SETQ SEMISTRING (CONCAT (ALLOCSTRING USE-SEMI-COLONS (CHARCODE ";")
                                                                " ")))
    else
      ; Balanced (hash bar) comment
      (SETQ SEMISTRING "#|")
      (SETQ SEMIWIDTH (STRINGWIDTH SEMISTRING FONT))
      (SETQ SEMICLOSE "#"))
[COND
  ((for C instring (PROGN
    STRING)
    ; dwimify bug tries to turn naked STRING into (STRING C) here.
    as I from 1 bind (POS _ (+ HERE (COND
      (P2FLG (CHARWIDTH (CHARCODE %)
                        FONT))
      ((NULL USE-SEMI-COLONS)
       0)
      (< USE-SEMI-COLONS 5)
      SEMIWIDTH)
      (T
       ; Include the width of the closing |#
       (TIMES 2 SEMIWIDTH)))
      CLOSEWIDTH))
    do [(COND
      ((EQ C (CHARCODE CR))
       ; Always want to print these strings specially
       (SETQ LASTSPACE I)
       (RETURN NIL))
      ((AND P2FLG (OR (EQ C (CHARCODE %)
                       (EQ C ESC)))
       ; Need escape
       (add POS ESCWIDTH)))
      (COND
       ((> (add POS (CHARWIDTH C FONT))
        RMARG)
        (RETURN NIL)))
      (COND
       ((EQ C (CHARCODE SPACE))
        (SETQ LASTSPACE I)
        finally (RETURN T))
       ; It all fits on this line
      (RETURN (COND
        (P2FLG (PRIN2S STRING TAIL STREAM))
        (T (if SEMISTRING
          then (PRIN1 SEMISTRING STREAM))
          (PRIN1S STRING TAIL STREAM)
          (if SEMICLOSE
           then (PRIN1 SEMICLOSE STREAM]
          (COND
            ((OR (NULL LASTSPACE)
              (AND (NULL COMMENTP)
                (NEQ HERE LMARG)
                MAKEMAP))
            ;; Can't print anything on this line before the end. Comments are allowed to have different first and subsequent margin in DEdit, but
            ;; not ordinary strings.
            (PRINENDLINE (SETQ HERE LMARG)
              STREAM)
            (SETQ LASTSPACE 0)))
          [COND
            (MAKEMAP
             ; Note start
             (SETQ MAPX1 HERE)
             (SETQ MAPY1 (DSPYPOSITION NIL STREAM))
             (SETQ SINGLELEFT (EQ HERE LMARG]
          [COND
            (P2FLG [COND
              ((NOT (IMAGESTREAMP STREAM))
               ; Need to be able to read it back
              (SETQ ESCAPESEPRS T)
              (LET ((HASH (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*)))
                (\OUTCHAR STREAM HASH)
                (add HERE (CHARWIDTH HASH FONT]
                (\OUTCHAR STREAM (CHARCODE %"))
                (add HERE (CHARWIDTH (CHARCODE %)
                                  FONT)))
              (USE-SEMI-COLONS
               ; Print the first set of semi-colons or #|
               (PRIN1 SEMISTRING STREAM)
               (add HERE SEMIWIDTH)
               (if (EQ USE-SEMI-COLONS 5)
                then
                  ; No more semis now
                  (SETQ SEMISTRING NIL]

```

;;; Now loop, printing as much as we can while there's room

```

(SETQ I 0)
LP [COND
  ([NULL (SETQ C (NTHCHARCODE STRING (add I 1)
    ; Done
    (GO DONE))
  (NOT (< I LASTSPACE))
  ; Must find the next safe place to print up to. LASTSPACE is either a space or CR position, or is 0, which is our state when printing

```

;;; Must find the next safe place to print up to. LASTSPACE is either a space or CR position, or is 0, which is our state when printing

```

;; from the left margin until we encounter a space.
(SETQ POS HERE)
(SETQ J I) ; Ordinarily, J is pointing at a space or CR except when we have
; just printed an endline

(SELCHARQ C
  (SPACE ; Would like all spaces before the eol, where they're invisible, not
; after
  (SELCHARQ (NTHCHARCODE STRING (ADD1 J))
    ((SPACE CR NIL)
      (SETQ LASTSPACE (ADD1 J)) ; Go ahead and print this space, and note that it is now okay to
; break the line
      (COND
        ((AND (>= (+ HERE SPACEWIDTH)
                  RMARG)
              (NOT ESCAPESEPRS)) ; Extra spaces have no effect, so don't print them at all, lest the
; dsprightmargin bite
          (GO LP))
        (T (GO PRINTIT))))
      NIL)
    (add POS SPACEWIDTH)) ; If two cr's in a row, print them all; if only one, must escape it
  (CR (COND
    ((EQ (SETQ C (NTHCHARCODE STRING (add I 1)))
        (CHARCODE CR))
      (PRINENDLINE LMARG STREAM)
      (while (EQ (SETQ C (NTHCHARCODE STRING (add I 1)))
                (CHARCODE CR))
        do (PRINENDLINE LMARG STREAM)))
      (ESCAPESEPRS (\OUTCHAR STREAM ESC)))
    (SETQ LASTSPACE 0)
    (GO ENDLINE))
    (PROGN ;; Gets set this way at left edge. Must print something on this line, even if there are no spaces before the right edge
      (GO CHECKESCAPE)))
    (SETQ LASTSPACE 0)
    (while (< POS RMARG) do (SELCHARQ (SETQ NEXTC (NTHCHARCODE STRING (add J 1)))
      ((CR SPACE) ; Can safely go this far
        (SETQ LASTSPACE J)
        (RETURN))
      (NIL ; End of string -- ok if there is space for closing quote and paren
        ; as well
        (COND
          ((< (+ POS CLOSEWIDTH)
              RMARG)
            (SETQ LASTSPACE J)
            (RETURN))
          (T (GO $$OUT))))
      NIL)
      (COND
        ((OR (EQ NEXTC (CHARCODE "%"))
            (EQ NEXTC ESC))
          (add POS ESCWIDTH)))
        (add POS (CHARWIDTH NEXTC FONT))
      finally (COND
        ((EQ LASTSPACE 0) ; Need break
          (COND
            [(EQ C (CHARCODE SPACE)) ; Will turn this space into CR
              (SETQ C (NTHCHARCODE STRING (add I 1))
                (T (SHOULDNT)))
              (GO ENDLINE]
          CHECKESCAPE
            (COND
              ((AND P2FLG (OR (EQ C (CHARCODE "%"))
                            (EQ C ESC))
                (\OUTCHAR STREAM ESC)
                (add HERE ESCWIDTH)))
            PRINTIT
              (\OUTCHAR STREAM C)
              (add HERE (CHARWIDTH C FONT))
              (GO LP)
            ENDLINE
              (PRINENDLINE LMARG STREAM)
              (SETQ HERE LMARG)
              (COND
                ((NULL C) ; Done
                  (GO DONE))
                ((AND ESCAPESEPRS (EQ (\SYNCODE SA C)
                                      SEPRCHAR.RC)) ; Have to quote sepr immediately following CR
                  (\OUTCHAR STREAM ESC)
                  (add HERE ESCWIDTH)
                  (GO PRINTIT))
                (T (COND
                    (SEMISTRING (PRIN1 SEMISTRING STREAM)
                      (add HERE SEMIWIDTH)))
                  (GO CHECKESCAPE)))
            DONE
              [COND

```

```

(P2FLG (\OUTCHAR STREAM (CHARCODE %"])
(COND
  [MAKEMAP (LET [(ENTRY (MAKEMAPENTRY TAIL (AND (NEQ MAKEMAP T)
                                                    MAKEMAP)
                                                    MAPX1 MAPY1 (DSPXPOSITION NIL STREAM)
                                                    (DSPYPOSITION NIL STREAM)
                                                    (\DEDITFONT# STREAM)
                                                    (replace LONGSTRINGP of ENTRY with T)
                                                    (COND
                                                     (SINGLELEFT (replace LONGSTRING1MARGINP of ENTRY with T)))
                                                     (COND
                                                      (EQ (- (DSPRIGHTMARGIN NIL STREAM)
                                                            LMARG)
                                                            RMARG)
                                                       ;; Assume that RMARG not equal to stream's right margin only happens for centered comments. In reality, it
                                                       ;; happens as well inside REPP, where RESETCLIP hides the true right margin.
                                                       (replace LONGSTRINGSYMMETRICP of ENTRY with T]
    (SEMICLOSE (PRIN1 SEMICLOSE STREAM)))
(RETURN])

```

)

(RPAQ? *DIVIDE-LONG-STRINGS* 'DISPLAY)

(PUTPROPS DSPRINTDEF COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1988 1990 1993))

FUNCTION INDEX

MAKEDOTPTAIL	3	PRIN2S	2	PRINENDLINE	2	SETFONT	3
PRIN1S	2	PRIN2STRING	4	PRINOPEN	2	SUPERPRINT/WRAPPER	4
PRIN2-LONG-STRING	4	PRINDOTP	3	PRINSHUT	2		

MACRO INDEX

BLANKS	1	OVERLAP	1	WIDTH	1	XPOSITION	1	YPOSITION	1
--------------	---	---------------	---	-------------	---	-----------------	---	-----------------	---

VARIABLE INDEX

DIVIDE-LONG-STRINGS	7
-----------------------------	---

RECORD INDEX

DEDITMAP	1
----------------	---

CONSTANT INDEX

DOTSTRING	1
-----------------	---
