File created: 18-Oct-93 16:47:22 **{Pele:mv:envos}<LispCore>Sources>CLTL2>NEWPRINTDEF.;2**

previous date: 3-Sep-91 18:15:55 {Pele:mv:envos}<LispCore>Sources>CLTL2>NEWPRINTDEF.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;

(RPAQQ **NEWPRINTDEFCOMS**
[(COMS

;;; A version of PRINTDEF abstracted so that it can be parameterized for non-teletype devices.

;;; One example is file DSPRINTDEF which provides one definition for the abstract fns such as WIDTH, XPOSITION etc used here.

```
        (FNS PRINTDEF SUPERPRINT SUPERPRINT0 SUBPRINT SUBPRINT1 PRINTPROG PRINTPROGVARS PRINTSQ BACKARROWP
             SUBPRINT/ENDLINE RPARS FITP DSFIT1 DSFIT2 SUPERPRINT/SPACE SUBPRINT/WRAPPERTAIL))
 [COMS                                                          ; Comment prettyprinter
        (FNS SUPERPRINT/COMMENT SEMI-COLON-COMMENT-P SUPERPRINT/COMMENT1 SUPERPRINT/COMMENT2)
        (INITVARS (COMMENTCOLUMN '(0.6 . 0.1))
               (*PRINT-SEMICOLON-COMMENTS* NIL)
               (*BACKQUOTE-WRAPPERS* '(BQUOTE %, %,@ %,.]
 (COMS                                                          ; Prettyprintmacros for common lisp and other poor things
        (FNS CODEWRAPPER.PRETTYPRINT PROG1.PRETTYPRINT CASE.PRETTYPRINT PROGV.PRETTYPRINT DO.PRETTYPRINT
             INDENTATION.FROM.HERE SEQUENTIAL.PRETTYPRINT)
        (ALISTS (PRETTYPRINTMACROS UNINTERRUPTABLY CL:UNWIND-PROTECT RESETLST CL:BLOCK CL:IF PROG1 CL:WHEN
                       CL:UNLESS WITH-READER-ENVIRONMENT CL:CATCH CASE CL:ECASE CL:ETYPECASE CL:TYPECASE
                       CL:PROGV WITH.MONITOR CL:DO* CL:DO CL:DOLIST CL:DOTIMES)
               (PRETTYEQUIVLST PROG* CL:COMPILER-LET)))
 [DECLARE%: EVAL@COMPILE DOCOPY (P (CL:PROCLAIM '(CL:SPECIAL DEFAULTFONT BOLDFONT USERFONT SYSTEMFONT
                                                  CLISPFONT BIGFONT PRETTYPRINTMACROS]
 (DECLARE%: EVAL@COMPILE DONTCOPY (GLOBALVARS CLISPARRAY CHANGESARRAY AVERAGEFNLENGTH %#CAREFULCOLUMNS
                                    AVERAGEVARLENGTH FONTWORDS FONTFNS CLISPCHARS FUNNYATOMLST
                                    PRETTYEQUIVLST COMMENTFLG *BACKQUOTE-WRAPPERS*)
        (BLOCKS (DSPRETTY PRINTDEF SUPERPRINT SUPERPRINT0 SUBPRINT SUBPRINT1 PRINTPROG PRINTPROGVARS
                       PRINTSQ BACKARROWP RPARS FITP DSFIT1 DSFIT2 (ENTRIES PRINTDEF SUPERPRINT FITP)
                       (SPECVARS TAIL LEFT FNSLST FIRSTPOS COMMENTCOL FORMFLG FILEFLG)
                       (LOCALFREEVARS TAILFLG CHANGEFLG)))
        (DECLARE%: DONTEVAL@LOAD (FILES (LOADCOMP)
                                        DSPRINTDEF])
```

;;; A version of PRINTDEF abstracted so that it can be parameterized for non-teletype devices.

;;; One example is file DSPRINTDEF which provides one definition for the abstract fns such as WIDTH, XPOSITION etc used here.

(DEFINEQ

(**PRINTDEF**

```
  [LAMBDA (EXPR LEFT FORMFLG TAILFLG FNSLST FILE)               ; Edited 15-Apr-88 11:59 by bvm
    (LET ((*STANDARD-OUTPUT* (GETSTREAM FILE 'OUTPUT))
          (MAKEMAP NIL)
          (%#RPARS (COND
                      ((AND %#RPARS (SYNTAXP (CHARCODE "]")
                                             'RIGHTBRACKET))    ; can only use brackets if read table supports them
                       %#RPARS)))
          SPACEWIDTH)
         (DECLARE (SPECVARS MAKEMAP SPACEWIDTH %#RPARS))
         (PROG [(FIRSTPOS (DSPLEFTMARGIN NIL *STANDARD-OUTPUT*))
                (RMARGIN (SUB1 (DSPRIGHTMARGIN NIL *STANDARD-OUTPUT*)))
                (TAIL (LIST EXPR))
                COMMENTCOL CHANGEFLG (FILEFLG (NEQ *STANDARD-OUTPUT* (TTYDISPLAYSTREAM]
               (DECLARE (SPECVARS RMARGIN FILEFLG FIRSTPOS))
               (COND
                  ((AND (NOT (IMAGESTREAMP *STANDARD-OUTPUT*))
                        (NOT FONTCHANGEFLG))
                   (DSPFONT 0 *STANDARD-OUTPUT*)))
               (RESETLST
                   (RESETSAVE NIL (LIST (FUNCTION DSPFONT)
                                        (DSPFONT NIL *STANDARD-OUTPUT*)
                                        *STANDARD-OUTPUT*))
                   (SETFONT DEFAULTFONT *STANDARD-OUTPUT*)
                   [COND
                      [PRETTYFLG [SETQ LEFT (COND
                                               ((NOT LEFT)
                                                FIRSTPOS)
                                               ((NUMBERP LEFT)
                                                (PLUS FIRSTPOS (BLANKS LEFT)))
                                               (T (DSPXPOSITION NIL *STANDARD-OUTPUT*]
```

```
                           (COND
                              ((GREATERP (DSPXPOSITION NIL *STANDARD-OUTPUT*)
                                      LEFT)
                               (TERPRI *STANDARD-OUTPUT*)))
                           (DSPXPOSITION LEFT *STANDARD-OUTPUT*)
                           (COND
                              (TAILFLG (SUBPRINT EXPR NIL NIL *STANDARD-OUTPUT*))
                              (T (SUPERPRINT EXPR TAIL NIL *STANDARD-OUTPUT*]
                     (T (COND
                           (TAILFLG (MAPRINT EXPR *STANDARD-OUTPUT* NIL NIL NIL (FUNCTION PRIN2S)))
                           (T (PRIN2S EXPR TAIL *STANDARD-OUTPUT*])])
```

(SUPERPRINT

```
  [LAMBDA (E TAIL BRFLG FILE)                                   ; Edited 14-Apr-88 18:44 by bvm
    (SETQ FILE (\GETSTREAM FILE 'OUTPUT))
    (COND
       [(NLISTP E)
        (OR [AND (NOT MAKEMAP)
                 (NOT (ATOM E))
                 (LET ((MACRO (ASSOC (TYPENAME E)
                                     PRETTYPRINTYPEMACROS)))
                      (AND MACRO (NEQ (APPLY* (CDR MACRO)
                                              E)
                                      E]
            (COND
               ((STRINGP E)
                (PRIN2STRING E TAIL FILE LEFT RMARGIN))
               (T (LET [(TEM (IDIFFERENCE RMARGIN (WIDTH E FILE T]
                                                                ; TEM is the last position at which E will fit
                        (COND
                           ((AND (ILESSP TEM (DSPXPOSITION NIL FILE))
                                 (IGREATERP TEM FIRSTPOS))
                            (SUBPRINT/ENDLINE (IMIN LEFT TEM)
                                   FILE)))
                        (PRIN2S E TAIL FILE]
       ((AND (SUPERPRINTEQ (CAR E)
                     COMMENTFLG)
             (OR FORMFLG (SEMI-COLON-COMMENT-P E)))
        (SUPERPRINT/COMMENT E FILE))
       ((AND PRETTYTRANFLG (NOT (ARGTYPE (CAR E)))
             (GETHASH E CLISPARRAY))
        (SUPERPRINT0 (GETHASH E CLISPARRAY)
              TAIL BRFLG FILE))
       (T (SUPERPRINT0 E TAIL BRFLG FILE])
```

(SUPERPRINT0

```
  [LAMBDA (E TAIL BRFLG FILE)                                   ; Edited 14-Apr-88 18:44 by bvm
                                                                ; BRFLG says do not print a ) as expression will be terminated by
                                                                ; a ].
    (PROG ((FN (CAR E))
           MACRO)
          [COND
             ((NOT (CL:SYMBOLP FN)))
             ((AND (SETQ MACRO (GET FN 'PRETTYWRAPPER))
                   (LISTP (CDR E))
                   (NULL (CDDR E))
                   (SETQ MACRO (CL:FUNCALL MACRO E FILE)))      ; Special case that DEDIT can handle: a 'wrapper' form wants to
                                                                ; pretty print via a read macro syntax
              (RETURN (SUPERPRINT/WRAPPER MACRO E TAIL BRFLG FILE)))
             ((SETQ MACRO (AND (NOT MAKEMAP)
                               (ASSOC FN PRETTYPRINTMACROS)))
              (COND
                 ((NOT (SETQ MACRO (APPLY* (CDR MACRO)
                                           E)))                 ; macro printed the thing
                  (RETURN E))
                 ((NEQ E MACRO)                                 ; macro returns something else to print (!)
                  (RETURN (SUPERPRINT MACRO TAIL BRFLG FILE)))
                 (T (SETQ E MACRO]
          (LET [(LEFT NIL)
                (NEWBR (AND (NULL BRFLG)
                            (FIXP %#RPARS)
                            (RPARS E %#RPARS]                   ; LEFT is set from within SUBPRINT.  Only appears here for call
                                                                ; to ENDLINE
               (PRINOPEN TAIL (COND
                                 (NEWBR '%[)
                                 (T '%()))
                         FILE)
               (SUBPRINT E (OR BRFLG NEWBR)
                      NIL FILE)
               [COND
                  ((ILESSP RMARGIN (IPLUS (DSPXPOSITION NIL FILE)
                                          (WIDTH ")" FILE)))
                   (PROG (TAIL)
                         ;; need to rebind tail because if next expression is a comment dont want to print it yet because we still have the right paren to
                         ;; print.
```

```
                    (SUBPRINT/ENDLINE LEFT FILE]
            (PRINSHUT TAIL (COND
                                (NEWBR '%])
                                (BRFLG NIL)
                                (T '%)))
                      FILE))
      (RETURN E])
```

(SUBPRINT

```
[LAMBDA (TAIL BRFLG END FILE)                                   ; Edited 26-Apr-88 10:48 by bvm
```

;; Prettyprint the elements of TAIL until we reach END.

```
(PROG [CURRENT DOCRFLG NEXT TEM OLDY CLISPWORD (FORMFLG FORMFLG)
            (FORMFLG0 FORMFLG)
            (TAIL0 TAIL)
            (LEFT0 (DSPXPOSITION NIL FILE))
            (CLW0 (CAR (SUPERPRINTGETPROP (CAR TAIL)
                              'CLISPWORD]
      (SETQ LEFT LEFT0)                                         ; LEFT is set from SUBPRINT.  Start where we are
  LP  [COND
         ((EQ TAIL END)
          (RETURN TAIL))
         ((NULL TAIL)
          (RETURN))
         ((NLISTP TAIL)
          (RETURN (PRINDOTP TAIL FILE]
      (SETQ OLDY (DSPYPOSITION NIL FILE))
      (SETQ CURRENT (CAR TAIL))
      [if (LITATOM CURRENT)
          then (if (AND (NEQ TAIL TAIL0)
                        (LISTP (CDR TAIL))
                        (NULL (CDDR TAIL))
                        (FMEMB CURRENT *BACKQUOTE-WRAPPERS*)
                        (SETQ TEM (GET CURRENT 'PRETTYWRAPPER))
                        (NEQ (CDR TAIL)
                             END)
                        (NOT MAKEMAP)
                        (SETQ TEM (CL:FUNCALL TEM TAIL FILE)))
                   then                                         ; tail of expression is something with a pretty wrapper, e.g., (foo .
                                                                ; ,bar), which if we printed it normally would come out (foo \, bar)
                        (SUBPRINT/WRAPPERTAIL TAIL TEM BRFLG)
                        (RETURN)
                 elseif [AND CLISPFLG FORMFLG0 (SETQ CLISPWORD (SUPERPRINTGETPROP CURRENT 'CLISPWORD]
                   then (OR (EQ CLW0 (CAR CLISPWORD))
                            (SETQ CLISPWORD NIL]
      [SETQ FORMFLG (AND FORMFLG0 (NOT (SUPERPRINTEQ (CAR TAIL0)
                                                     'QUOTE]
```

;; says whether next expression is to be treated as a form. used to be an argument to superprint but this value of formflg should also affect the
;; call to endline from subprint.

```
      (SETFONT (PROG1 (AND FORMFLG0 (LITATOM CURRENT)
                           (SETFONT (COND
                                       ((LISTP CLISPWORD)
                                        CLISPFONT)
                                       ((FMEMB CURRENT FONTWORDS)
                                        USERFONT)
                                       ((AND (EQ TAIL0 TAIL)
                                             (NULL END))
                                        (COND
                                           ((OR (FMEMB CURRENT FNSLST)
                                                (FMEMB CURRENT (LISTP FONTFNS)))
                                            USERFONT)
                                           ((FGETD CURRENT)
                                            SYSTEMFONT)))
                                       ((AND (SUPERPRINTGETPROP CURRENT 'CLISPTYPE)
                                             (NOT (FMEMB CURRENT CLISPCHARS)))
                                                                ; Infix operators like GT AND etc.
                                        CLISPFONT))
                                    FILE))
                      ;; When printing a function via a call to prettydef and fontflg is turned on and the function is either on FNS or on
                      ;; FONTFLG do a fontchange.
                      (SETQ CURRENT (SUPERPRINT CURRENT TAIL (AND (NULL (CDR TAIL))
                                                                  BRFLG)
                                                FILE)))
            FILE)                                               ; Reason for (SETQ CURRENT --) is in case CURRENT is
                                                                ; printed as something else
```

;; Popping TAIL used to be done in the call to SUPERPRINT. But this can cause subsequent comments to be printed first if ENDLINE is called
;; because of no space. BRFLG only affects last expression in list.

```
      (SETQ TAIL (CDR TAIL))
```

;; CURRENT is always the element just printed; NEXT the one about to be i.e. CAR of TAIL

```
 LP0 (COND
        ((OR (EQ TAIL END)
```

```
                (NLISTP TAIL))
            (GO LP))
           ((OR (NULL CLISPFLG)
                (NULL FORMFLG)
                (NULL FORMFLG0))                                ; Skip this clisp stuff
            (GO LP1))
           [(NOT (LITATOM (SETQ NEXT (CAR TAIL]
           ([AND (SETQ TEM (SUPERPRINTGETPROP NEXT 'CLISPWORD))
                 (OR (NLISTP TEM)
                     (EQ CLW0 (CAR TEM]
            ;; AND and OR are treated like prettywords because they are broadscope operators i.e. they permit segments and therefore the
            ;; standard FITP test can't be used.
            (GO CLISPWORD))
           ([AND (EQ (CADR (LISTP TAIL))
                     '_)
                 (OR (SUPERPRINTEQ (CAR TAIL0)
                           'CREATE)
                     (SUPERPRINTEQ (CAR TAIL0)
                           'create]
            (GO CR)))
         (COND
           ((LISTP CURRENT))
           ((NOT (LITATOM CURRENT))
            (GO LP1))
           ((SELECTQ (CAR CLISPWORD)
                 ((IFWORD FORWORD)
                     T)
                 NIL)
            (SETQ DOCRFLG NIL)
            (COND
               ((NULL END)
                (SETQ END T)))                                  ; See use of END below

            )
           ([NOT (OR (NULL CLISPIFYPACKFLG)
                     (ATOM NEXT)
                     (COND
                        [(EQ TAIL (CDR TAIL0))
                         (OR (FGETD CURRENT)
                             (SUPERPRINTGETPROP CURRENT 'EXPR]
                        (T (BOUNDP CURRENT)))
                     (FMEMB CURRENT FUNNYATOMLST)
                     (NOT (FMEMB (SETQ TEM (NTHCHAR CURRENT -1))
                                 CLISPCHARS))
                     (EQ TEM '>]                                ; E.g.  X* (FOO) Don't space
            (GO LP))
           ((BACKARROWP CURRENT)                                ; E.G.  IF -- THEN FOO_X FIE_Y is more readable if the
                                                                ; assignments are on separate lines.
            (GO CR)))
       LP1 [COND
           ((EQ TAIL (CDR TAIL0))                               ; First time through i.e. just superprinted HEAD of list.
            (COND
               ((LISTP CURRENT)
                (GO CR))
               ((AND FORMFLG0 (SELECTQ (OR (CDR (FASSOC CURRENT PRETTYEQUIVLST))
                                           CURRENT)
                                       (COND (SETQ LEFT (IPLUS LEFT0 (WIDTH "CO" FILE)))
                                             (GO CR))
                                       ((PROG RESETVARS)
                                            (RETURN (PRINTPROG TAIL BRFLG FILE CURRENT)))
                                       (SELECTQ (RETURN (PRINTSQ TAIL BRFLG FILE)))
                                       ((SETQ RESETVAR)
                                            (GO SP))
                                       (FUNCTION                ; If FUNCTION has a second arg, fall thru and reset margin.
                                                                ; Else leave it for compactness
                                                 (OR (CDR TAIL)
                                                     (GO SP)))
                                       ([LAMBDA NLAMBDA]
                                            (SETQ DOCRFLG T)
                                            (SETQ LEFT (IPLUS LEFT0 (BLANKS 1)))
                                            (SUPERPRINT/SPACE FILE)
                                            (GO LP))
                                       NIL)))
               ((NOT (FITP TAIL T [OR (LISTP END)
                                      (AND CLISPWORD (SUBPRINT1 TAIL (CAR CLISPWORD]
                          NIL FILE))
                (GO CR)                                         ; Don't reset I.

                )
               (T (SUPERPRINT/SPACE FILE)                       ; Default head of form handling
                  [SETQ LEFT (IMIN (DSPXPOSITION NIL FILE)
                                   (IPLUS LEFT0 (BLANKS 6]       ; Dont indent too far
                  (GO LP]
         (COND
           ([AND (NEQ OLDY (DSPYPOSITION NIL FILE))
                 (OR (NOT (ATOM CURRENT))
```

```
                    (EQ CURRENT '>]
          (GO CR)))
```

;; Printing last 'thing' (usually a list) caused a c.r. Also occurs if printing angle brackets which contain a list inside e.g. < (FOO (FIE) X) > and c.r. ;; will occur after >.

```
      (SETQ NEXT (CAR TAIL))
      (COND
         [(LISTP CURRENT)
          (COND
             ((OR (NULL END)
                  (SUPERPRINTEQ (CAR CURRENT)
                        COMMENTFLG))
              (GO CR))
             ((AND (LISTP NEXT)
                   (SUPERPRINTEQ (CAR NEXT)
                         COMMENTFLG))
              (GO SP))
             ([AND (LITATOM NEXT)
                   (OR (SUPERPRINTGETPROP NEXT 'CLISPWORD)
                       (SUPERPRINTGETPROP NEXT 'CLISPTYPE]
              (GO SP))
             (T (GO CR]
         ((NLISTP NEXT)
          (GO SP))
         (DOCRFLG
```

;; DOCRFLG is set to T whenever a carriage return is performed. It is reset to NIL whenever a carriage return is NOT ;; performed e.g. when two atoms are adjacent. while it is T carriage returns are performed FOLLOWING all expressions. ;; For example in (A B (C) D (E) F G (H)) (C) D (E) and F would be on separate lines but F G and (H) would all be on the ;; same line.

```
                (GO CR))
         ((FITP NEXT NIL NIL NIL FILE)
          (GO SP))
         (T (GO CR)))
   SP    (SETQ DOCRFLG NIL)
         (SUPERPRINT/SPACE FILE)
         (GO LP)
   CR    (SETQ DOCRFLG T)
         (SUBPRINT/ENDLINE NIL FILE)
         (GO LP)
   CLISPWORD
         (PROG ((LEFT LEFT)
                (LEFT0 LEFT0)
                (CEND))
               (SELECTQ (OR (CDR (FASSOC NEXT PRETTYEQUIVLST))
                            NEXT)
                    ((THEN ELSE ELSEIF then else elseif)        ; THEN ELSE and ELSEIF always start a new line.
                         (SETQ LEFT (IPLUS (SUBPRINT/ENDLINE (IPLUS LEFT0 (BLANKS (SELECTQ NEXT
                                                                                    ((THEN then)
                                                                                       3)
                                                                                    1)))
                                                             FILE)
                                           (BLANKS 1)))
```

;; Note that in most cases LEFT will be reset again in subprint after printing the CLISPWORD. It will remain this value ;; only if the next expression wont fit.

```
                         (SETQ TAIL (SUBPRINT TAIL BRFLG (SUBPRINT1 (CDR TAIL)
                                                                    'IFWORD END)
                                                    FILE))
                         (RETURN))
                    ((AND OR and or)                            ; So when new left margin is coputed in next cond it will be
                                                                ; based on inner expression.
                         (SETQ LEFT0 LEFT)
                         (SETQ CEND (SUBPRINT1 (CDR TAIL)
                                               NIL END)))
                    ((! !!)
                         (SETQ CEND (CDDR TAIL)))
                    (SETQ CEND (SUBPRINT1 (CDR TAIL)
                                          (CAR (GETP (CAR TAIL0)
                                                     'CLISPWORD))
                                          END)))
               (SETQ LEFT (IPLUS (COND
                                   ((AND (EQ OLDY (DSPYPOSITION NIL FILE))
                                         (FITP TAIL NIL CEND NIL FILE))
                                    (SUPERPRINT/SPACE FILE)
                                    (DSPXPOSITION NIL FILE))
                                   (T
```

;; Either last expression involved a CR e.g. FOR X IN (FOO (FIE) (FUM)) DO -- OR the segment of ;; the list between here and the next CLISPFORWORD will not fit.

```
                                      (SUBPRINT/ENDLINE (IPLUS LEFT0 (BLANKS 2))
                                                        FILE)))
                                 (BLANKS 1)))
               (SETQ OLDY (DSPYPOSITION NIL FILE))
               (SETQ CURRENT (CAR (NLEFT TAIL 1 CEND)))
               (SETQ TAIL (SUBPRINT TAIL BRFLG CEND FILE)))
      (GO LP0)
```

```
      ;; We are now in the position of just having printed the element before E and are ready to look ahead at the next one so go to LP0.

       ])
```

(SUBPRINT1

```
  [LAMBDA (LST X END)                                                    (* bas%: "24-NOV-81 15:28")
    (bind TMP for L on LST until [OR (EQ L END)
                                     [AND (LITATOM (CAR L))
                                          (SETQ TMP (GETPROP (CAR L)
                                                             'CLISPWORD))
                                          (OR (NULL X)
                                              (EQ X (CAR TMP]
                                     (AND (EQ X 'RECORDWORD)
                                          (EQ (CADR L)
                                              '_]
       finally (RETURN L])
```

(PRINTPROG

```
  [LAMBDA (TAIL BRFLG FILE PROGWORD)                                     ; Edited 14-Apr-88 18:44 by bvm
    (PROG [(LABELL (IDIFFERENCE (DSPXPOSITION NIL FILE)
                                (STRINGWIDTH "ROG" FILE)))
           (FORMLEFT (IPLUS (DSPXPOSITION NIL FILE)
                            (STRINGWIDTH " " FILE]                       ; LABELL is the position PROG labels start in;  FORMLEFT that
                                                                         ; for forms
          (DSPXPOSITION FORMLEFT FILE)
          (COND
             ((AND (CAR TAIL)
                   (LITATOM (CAR TAIL)))
              (SUPERPRINT (CAR TAIL)
                     TAIL
                     (PROGN (SETQ TAIL (CDR TAIL))
                            T)
                     FILE)
              (SPACES 1 FILE)))                                          ; Print PROG variables.
          (PRINTPROGVARS TAIL FILE (AND (NULL (SETQ TAIL (CDR TAIL)))
                                        BRFLG))
      LP1 (COND
             ((LISTP TAIL)
              (SUBPRINT/ENDLINE LABELL FILE)))                           ; ENDLINE resets TAIL when it sees a comment.
      LP2 (COND
             ((NLISTP TAIL)
              (AND TAIL (PRINDOTP TAIL FILE))
              (RETURN))
             ((LISTP (CAR TAIL))
              (COND
                 ((ILEQ FORMLEFT (DSPXPOSITION NIL FILE))
                  (PRINENDLINE FORMLEFT FILE))
                 (T (DSPXPOSITION FORMLEFT FILE)))
              (SUPERPRINT (CAR TAIL)
                     TAIL
                     (AND (NULL (SETQ TAIL (CDR TAIL)))
                          BRFLG)
                     FILE)
              (GO LP1))
             (T (COND
                   ((ILESSP LABELL (DSPXPOSITION NIL FILE))              ; Two labels in a row
                    (PRINENDLINE LABELL FILE)))
                (SUPERPRINT (CAR TAIL)
                       TAIL NIL FILE)                                    ; Print the label.
                (pop TAIL)
                (GO LP2])
```

(PRINTPROGVARS

```
  [LAMBDA (TAIL FILE BRFLG)                                              (* bvm%: " 4-May-86 15:01")
```

;;; (CAR TAIL) is a VARS list for a PROG etc. Print it suitably

```
    (SUPERPRINT (CAR TAIL)
           TAIL BRFLG FILE])
```

(PRINTSQ

```
  [LAMBDA (TAIL BRFLG FILE)                                              (* bvm%: " 2-Jun-86 15:07")
    (PROG ((KEYL (QUOTIENT (PLUS LEFT (DSPXPOSITION NIL FILE))
                           2))
           FOLD LEFT REST)                                               ; KEYL is the position keys start in;  LEFT that for forms
                                                                         ; Print select expression FORMFLG=T
          (SUPERPRINT/SPACE FILE)
          [SETQ FOLD (IPLUS (SETQ LEFT (DSPXPOSITION NIL FILE))
                            (TIMES 2 (DIFFERENCE LEFT KEYL]
          (SUPERPRINT (CAR TAIL)
                 TAIL NIL FILE)
      LP  (OR (SETQ TAIL (CDR TAIL))
              (RETURN))
```

```
          (PRINENDLINE KEYL FILE)
          (COND
             ((NLISTP TAIL)
              (RETURN (PRINDOTP TAIL FILE)))
             [(CDR TAIL)
              (COND
                 ((LISTP (CAR TAIL))
                  (PRINOPEN TAIL '%( FILE)
                  (PROG (FORMFLG)                                   ; Print keys not as function
                        (SUPERPRINT (CAAR TAIL)
                                (CAR TAIL)
                               NIL FILE))
                  (AND (SETQ REST (CDAR TAIL))
                       (PROG ((LEFT LEFT)
                              (HERE (DSPXPOSITION NIL FILE)))
                             (SUPERPRINT/SPACE FILE)
                             (COND
                                ((OR (LISTP (CAAR TAIL))
                                     (IGEQ HERE FOLD))
                                 [COND
                                    ((AND (LISTP (CAR REST))
                                          (EQMEMB (CAAR REST)
                                                  COMMENTFLG))      ; Start comment on same line
                                     (PROG ((LEFT LEFT))
                                           (SUBPRINT REST NIL (CDR REST)
                                                  FILE))
                                     (SETQ REST (CDR REST]
                                 (PRINENDLINE LEFT FILE))
                                (T (SETQ LEFT HERE)))
                             (SUBPRINT REST NIL NIL FILE)))
                  (PRINSHUT TAIL '%) FILE))
                 (T (PRIN2S (CAR TAIL)
                            TAIL FILE]
             (T (SUPERPRINT (CAR TAIL)
                        TAIL BRFLG FILE)))
          (GO LP])
```

(BACKARROWP

```
  [LAMBDA (X)                                                       (* bas%: "17-NOV-82 15:19")
    (AND (STRPOS '_ X)
         (NEQ (NTHCHARCODE X -1)
              (CHARCODE _])
```

(SUBPRINT/ENDLINE

```
  [LAMBDA (N FILE)                                                  (* lmm "30-Jul-85 03:20")
    (AND FORMFLG (while (SUPERPRINTEQ [CAR (LISTP (CAR (LISTP TAIL]
                              COMMENTFLG)
                    do (SUPERPRINT (CAR TAIL)
                              TAIL NIL FILE)                        ; a comment
                       (pop TAIL)))
    (PRINENDLINE (OR N LEFT)
          FILE)
    N])
```

(RPARS

```
  [LAMBDA (E NP)                                                    (* bas%: "11-MAR-83 11:45")
    (COND
       ((ILEQ NP 0))
       ((NLISTP E)
        NIL)
       (T (SELECTQ (CAR E)
                   ([LAMBDA NLAMBDA]
                       T)
                   (DEFINEQ ;; Dont want square brakcets around DEFINEQ expressions, because this means last function pair is special with
                            ;; respect to LOADFNS
                            NIL)
                   (RPARS (CAR (LAST E))
                          (SUB1 NP])
```

(FITP

```
  [LAMBDA (X TAILFLG ENDTAIL LSTCOL FILE)                           ; Edited 14-Apr-88 18:39 by bvm
```

;; Value is T unless X doesnt fit. There are two cases: one where X is a tail (only called for the first tail i.e. CDR of an expression) and the second ;; where it is an element. They differ in their treatment of linear lists of atoms. If one is about to print (FOO A B C D E F) and it wont fit on a line ;; then do a carriage return and start printing. However if A B C D E F doesnt fit doesnt mean to do a carriage return (and then line all the atoms up ;; in a column). The idea is that long lists are given as much room as possible (the first carriage return) but not at the expense of making them be ;; vertical.

```
    (DECLARE (SPECVARS ENDTAIL))                                    ; ENDTAIL is the end of TAIL e.g. when printing CLISP
                                                                    ; segments
    (OR FILE (SETQ FILE *STANDARD-OUTPUT*))                         ; Don't let FILE be NIL, since CHARWIDTH and STRINGWIDTH
                                                                    ; won't default correctly
    (LET* [(SPACEWIDTH (CHARWIDTH (CHARCODE SPACE)
```

```
                              FILE))
            (CAREFUL (BLANKS %#CAREFULCOLUMNS))
            (N (- (OR LSTCOL RMARGIN)
                  (DSPXPOSITION NIL FILE]
           (DECLARE (SPECVARS SPACEWIDTH CAREFUL))
           (COND
              (TAILFLG (AND (> N (BLANKS (+ AVERAGEVARLENGTH 2)))
                            (DSFIT1 X N NIL FILE)))
              (T (DSFIT2 X N NIL FILE])
```

(DSFIT1

```
[LAMBDA (LST N N1 FILE)                                    (* lmm "30-Jul-85 03:08")
  (DECLARE (USEDFREE CAREFUL ENDTAIL))
```

;; Checks to see if LST could fit in N spaces.

```
  (bind (M _ (COND
                (TAILFLG NIL)
                (T N)))
     for L on LST until (EQ L ENDTAIL) do (COND
                                             [(NLISTP (CAR L))
                                              (COND
                                                 (M [SETQ M (IDIFFERENCE M (IPLUS (COND
                                                                                     ((ILESSP M CAREFUL)
```

; When getting near right margin actually perform the WIDTH
; check.

```
                                                                                      (WIDTH (CAR L)
                                                                                             FILE T))
                                                                                     (T (BLANKS AVERAGEVARLENGTH
                                                                                               )))
                                                                                  (BLANKS 1]
                                                    (COND
                                                       ((ILESSP M 0)
                                                        (RETURN NIL]
                                             ((DSFIT2 (CAR L)
                                                      (OR N1 N)
                                                      NIL FILE)
```

;; The extra argument to DSFIT1 is for use in connectionwith CLISPPRETTYWORDS e.g. FOR IF
;; etc. Normally we figure that any lists can be printed at the position corresponding to the first
;; argument ut with FOR's and IF's et al they would always be preceded by the corresponding CLISP
;; word.

```
                                                       (AND M (SETQ M N)) ; Reset count when LISTP reached as margin will be reset

                                                       )
                                             (T (RETURN NIL)))
     finally (RETURN T])
```

(DSFIT2

```
[LAMBDA (X N NC FILE)                                      (* lmm "30-Jul-85 03:09")
  (DECLARE (USEDFREE CAREFUL))                             ; NC is local to DSFIT2
  (COND
     ((SUPERPRINTEQ (CAR X)
            COMMENTFLG)
      T)
     [(LISTP (CAR X))                                      ; Non-atomic CAR of form e.g. COND clause open lambda etc.
      (AND [ILESSP 0 (SETQ N (IDIFFERENCE N (WIDTH "()" FILE]
           (DSFIT2 (CAR X)
                   N NIL FILE)
           (OR (NULL (CDR X))
               (DSFIT1 (CDR X)
                       N NIL FILE]
     ([ILESSP N (IPLUS (WIDTH "()" FILE)
                       (SETQ NC (COND
                                   ((ILESSP N CAREFUL)
                                    (WIDTH (CAR X)
                                           FILE T))
                                   (T (BLANKS AVERAGEFNLENGTH]
```

;; Checks to see if there is space for function name and two parentheses. when there are more than CAREFUL columns left approximate
;; using AVERAGEFNLENGTH.

```
      NIL)
     ((NULL (CDR X))
      T)
     ([ILEQ [SELECTQ (CAR X)
                 (COND 0)
                 (FUNCTION (WIDTH "(FUNCTION LAMBDA ABC)" FILE))
                 ([LAMBDA NLAMBDA]
                      (WIDTH "(LAMBDA ABC" FILE))
                 (SETQ (IPLUS (WIDTH "(SETQ " FILE)
                              (BLANKS AVERAGEVARLENGTH)))
                 (PROGN (SETQ N (IDIFFERENCE N NC))
                        (BLANKS (ADD1 AVERAGEFNLENGTH]
            (SETQ N (IDIFFERENCE N (BLANKS 2]
```

;; The two spaces correspond to the amount LEFT would be decremented on the recursive call to superprint. the default clause in the selectq
;; checks to see if function and at least one atomic argument (we know there is at least one) will fit. The call to DSFIT1 checks to see if using

```
        ;; normal alignment algorithm the expression can fit.
        (DSFIT1 (CDR X)
               N
               (SELECTQ (CAR (SUPERPRINTGETPROP (CAR X)
                                    'CLISPWORD))
                    ((IFWORD FORWORD)
                         (IDIFFERENCE N (IPLUS NC (BLANKS 1))))
                    NIL)
               FILE])
```

(SUPERPRINT/SPACE

```
  [LAMBDA (FILE)                                          ; Edited 31-Mar-88 12:18 by bvm
    ;; Print a space, preparing for next item to be printed
    (DECLARE (CL:SPECIAL RMARGIN SPACEWIDTH LEFT))        ; bound by prettyprinter stuff
    (if (< (- RMARGIN (DSPXPOSITION NIL FILE))
           (TIMES 2 SPACEWIDTH))
        then                                              ; printing a space will overflow the line, or if not then the next
                                                          ; char would, so go to new line
          (PRINENDLINE LEFT FILE)
      else (PRIN3 " " FILE])
```

(SUBPRINT/WRAPPERTAIL

```
  [LAMBDA (TAIL MACRO BRFLG)                              ; Edited 15-Apr-88 11:54 by bvm
    ;; Called when TAIL = ... wrapperform body), e.g, QUOTE FOO).  We print this as a dotted tail with the wrapper instead.
    (LET ((DOT ". ")
          (BODY (CADR TAIL)))
         (if (NOT (if (NLISTP BODY)
                      then (< (+ (DSPXPOSITION)
                                 (WIDTH DOT)
                                 (WIDTH MACRO)
                                 (WIDTH BODY NIL T)
                                 (WIDTH ")"))
                              RMARGIN)
                  else (FITP BODY)))
             then                                         ; Start a new line
                  (PRINENDLINE LEFT))
         (PRIN3 DOT)
         (PRIN3 MACRO)
         (SUPERPRINT BODY (CDR TAIL)
                BRFLG *STANDARD-OUTPUT*])

)

;; Comment prettyprinter

(DEFINEQ

(SUPERPRINT/COMMENT
  [LAMBDA (L FILE)                                        ; Edited 13-Apr-88 12:55 by bvm
    (DECLARE (USEDFREE LEFT TAIL RMARGIN FILEFLG MAKEMAP))
    (COND
       ((AND **COMMENT**FLG (NOT FILEFLG)
             (NOT MAKEMAP))
        ;; If we're eliding comments, not printing to a file, and not in DEdit, then just print the elision string
        (COND
           ((> (+ (DSPXPOSITION NIL FILE)
                  (STRINGWIDTH **COMMENT**FLG FILE))
               (DSPRIGHTMARGIN NIL FILE))                 ; Watch out for overflowing the current line.
            (PRINENDLINE (DSPLEFTMARGIN NIL FILE)
                   FILE)))
        (PRIN1S **COMMENT**FLG NIL FILE))
       (T (PROG ((DSLMARG (DSPLEFTMARGIN NIL FILE))
                 (HERE (DSPXPOSITION NIL FILE))
                 (COMMENT-RMARGIN RMARGIN)
                 (SEMIP (SEMI-COLON-COMMENT-P L))
                 COMMENT-LMARGIN RIGHTFLG BODY HALFLINE)
                [if SEMIP
                    then                                  ; Extract the comment body
                         (COND
                            ((OR [NOT (STRINGP (SETQ BODY (CAR (LISTP (CDR (LISTP (CDR L]
                                 (CDDDR L))               ; Not a good semi-colon comment
                             (SETQ SEMIP NIL]
                [COND
                   [[SETQ RIGHTFLG (if SEMIP
                                       then               ; Only 1-semi comments go in right margin
                                            (EQ SEMIP 1)
                                     else                 ; Short single * comments go at right
                                          (AND (NOT (SUPERPRINTEQ (CADR L)
                                                                  COMMENTFLG))
                                               (<= (LENGTH L)
                                                   15]    ; Print comment in the righthand margin
```

```
        (SETQ COMMENT-LMARGIN (OR COMMENTCOL (SUPERPRINT/COMMENT1 L RMARGIN FILE]
       ((AND SEMIP (NOT MAKEMAP))                    ; Semi-colon comment > 1, unless under DEdit (lest we confuse
                                                     ; it)
        (AND SEMIP (> SEMIP 2)
             (NOT MAKEMAP))
        (SETQ COMMENT-LMARGIN (if (EQ SEMIP 2)
                                  then               ; indent like code, but no more than a third of the way over if it
                                                     ; would take more than 2 lines to print this.
                                       [MIN LEFT (MAX (- RMARGIN (FIXR (TIMES (STRINGWIDTH BODY FILE
                                                                              )
                                                                       0.52)))
                                                      (+ DSLMARG (IQUOTIENT (- RMARGIN DSLMARG)
                                                                            3]
                                  else               ; Comment should be printed flush left.
                                     DSLMARG)))
       (T (LET ((INDENT (IQUOTIENT (- RMARGIN DSLMARG)
                                   11)))             ; Print old-style comment centered and wide, indented about
                                                     ; 10% from margins
               (SETQ COMMENT-LMARGIN (+ DSLMARG INDENT))
               (SETQ COMMENT-RMARGIN (- RMARGIN INDENT))
               (COND
                  ((EQ HERE COMMENT-LMARGIN)
                   ;; HACK: Almost certainly called from REPP, so we must supress the normal leading and trailing blank lines as
                   ;; they have already been done
                   (SETQ RIGHTFLG T]
    (COND
       ((AND (NULL RIGHTFLG)
             (OR (NOT SEMIP)
                 (> SEMIP 1)))                       ; Centered comment starts on new line
        (if (> HERE COMMENT-LMARGIN)
            then                                     ; We have not yet moved down a line, so do that first
                 (TERPRI FILE))
        [if (AND (EQ SEMIP 2)
                 (IMAGESTREAMP FILE))
            then                                     ; For 2-semi comments, only go down half line, accomplished by
                                                     ; moving up half line now before this next endline
                 (RELMOVETO 0 (SETQ HALFLINE (IQUOTIENT (- (DSPLINEFEED NIL FILE))
                                                        2]
        (PRINENDLINE COMMENT-LMARGIN FILE))
       ((< COMMENT-LMARGIN (DSPXPOSITION NIL FILE))  ; Past the starting point, so start new line
        (PRINENDLINE COMMENT-LMARGIN FILE))
       (T (DSPXPOSITION COMMENT-LMARGIN FILE)))
    (SETFONT (PROG1 (SETFONT COMMENTFONT FILE)
                    (COND
                       ((AND SEMIP (NOT MAKEMAP)
                             (OR *PRINT-SEMICOLON-COMMENTS* (IMAGESTREAMP FILE)))
                                                     ; do nice semi-colon stuff
                        (PRIN2-LONG-STRING BODY FILE NIL NIL COMMENT-LMARGIN COMMENT-RMARGIN T SEMIP))
                       (T                            ; Old comment or in DEdit (makemap true), so have to do it the
                                                     ; old way
                          (SETQ SEMIP NIL)
                          (SUPERPRINT/COMMENT2 L COMMENT-LMARGIN (IQUOTIENT (+ COMMENT-LMARGIN
                                                                               COMMENT-RMARGIN)
                                                                            2)
                                               COMMENT-RMARGIN FILE))))
            FILE)
    (if (OR (NULL SEMIP)
            (> SEMIP 2))
        then                                         ; Old centered comments and big semi-colon comments get new
                                                     ; line
             (OR RIGHTFLG (PRINENDLINE DSLMARG FILE))
      elseif (NULL (CDR TAIL))
        then                                         ; Nothing more will be printed.  So even though we were a short
                                                     ; comment, we need to go to new line so that the closing paren is
                                                     ; on a new line, rather than here after the comment (AR 8475)
             (PRINENDLINE LEFT FILE)
      elseif [AND HALFLINE (NOT (AND (LISTP (CDR TAIL))
                                     (SEMI-COLON-COMMENT-P (LISTP (CADR TAIL]
        then                                         ; Set off double-semi-colon comment by half line.  Don't do for
                                                     ; consecutive comments, since the next comment will take care
                                                     ; of it
             (RELMOVETO 0 HALFLINE)
             (PRINENDLINE DSLMARG FILE))
    (RETURN L])
```

(SEMI-COLON-COMMENT-P

```
  [LAMBDA (E)                                        ; Edited 20-Sep-87 18:30 by raf
    ;; If E is a comment, returns a number giving number of semis (or type).
    (SELECTQ (CADR E)
         (;                                          ; SEdit-style right-margin comment
           1)
         (;;                                         ; SEdit-style current-indent comment
            2)
         (;;;                                        ; SEdit-style flush left comment
```

```
          3)
        (;;;;                                                    ; Page boundary type comment
             4)
        (%|                                                      ; Balanced (hash vertical bar) comment
           5)
        NIL])
```

(SUPERPRINT/COMMENT1

```
  [LAMBDA (CF RMARGIN FILE)                                      (* bvm%: "26-Mar-86 14:03")
```

;;; Computes the left margin for comments printed on the right

```
    (LET ((EDITDATEP (EDITDATE? CF))
          LM MINLEFT DEFAULT)
         (SETQ MINLEFT (IDIFFERENCE [IDIFFERENCE RMARGIN (COND
                                                            [EDITDATEP
                                                                 ; Min space is size of this edit date comment
                                                               (LET ((FONT (DSPFONT COMMENTFONT FILE)))
                                                                    (PROG1 (WIDTH CF FILE T)
                                                                           (DSPFONT FONT FILE]
                                                            (T   ; Else an arbitrary space
                                                               (BLANKS 15]
                                    (BLANKS 1)))
         (SETQ DEFAULT (FIXR (TIMES (OR (FLOATP (CAR (LISTP COMMENTCOLUMN)))
                                        0.6)
                                    RMARGIN)))
         (SETQ LM (IMAX (IQUOTIENT RMARGIN 2)
                        (IMIN MINLEFT DEFAULT)))                 ; use at least enough space, but no more than half the line
         (COND
            ((NOT EDITDATEP)                                     ; Don't have the editdate dictate margin for rest of function!
             (SETQ COMMENTCOL LM)))
         LM])
```

(SUPERPRINT/COMMENT2

```
  [LAMBDA (CMT COMMENT-LMARGIN COMMENT-MIDPOINT COMMENT-RMARGIN FILE)
                                                                 (* bvm%: "28-May-86 15:15")
    (SETQ FILE (\GETSTREAM FILE 'OUTPUT))
    (PRINOPEN TAIL '%( FILE)
    (for TAIL on CMT bind LASTITEM THISITEM
       do (SETQ THISITEM (CAR TAIL))                             ; Decide whether to continue on a new line
          (COND
             ([OR (EQ LASTITEM '-)
                  [AND (IGEQ (DSPXPOSITION NIL FILE)
                             COMMENT-MIDPOINT)
                       (OR (LISTP THISITEM)
                           (AND (LITATOM LASTITEM)
                                (SELCHARQ (NTHCHARCODE LASTITEM -1)
                                     ((; %. -)
                                      T)
                                     NIL]
                  (PROGN (COND
                            ((AND (NEQ TAIL CMT)
                                  (OR (NLISTP LASTITEM)
                                      (SELECTQ THISITEM
                                           ((%. %, ; %:)
                                            NIL)
                                           T)))                  ; Space before next element unless it looks like punctuation after
                                                                 ; a list
                             (SUPERPRINT/SPACE FILE)))
                         (AND (NLISTP THISITEM)
                              (NOT (STRINGP THISITEM))
                              (IGEQ (IPLUS (DSPXPOSITION NIL FILE)
                                           (WIDTH THISITEM FILE T)
                                           (WIDTH (COND
                                                     ((CDR TAIL)
                                                      " ")
                                                     (T          ; Leave space for the paren; i.e., don't print last atom on one line
                                                                 ; and the paren on the next
                                                        ")"))
                                                  FILE))
                                    COMMENT-RMARGIN]
              (PRINENDLINE COMMENT-LMARGIN FILE)))
          (COND
             ((LISTP (SETQ LASTITEM THISITEM))
              (SUPERPRINT/COMMENT2 LASTITEM COMMENT-LMARGIN COMMENT-MIDPOINT COMMENT-RMARGIN FILE))
             ((STRINGP LASTITEM)
              (PRIN2STRING LASTITEM TAIL FILE COMMENT-LMARGIN COMMENT-RMARGIN T))
             (T (PRIN2S LASTITEM TAIL FILE)))
       finally (AND TAIL (PRINDOTP TAIL FILE)))
    (PRINSHUT TAIL '%) FILE])

)

(RPAQ? COMMENTCOLUMN '(0.6 . 0.1))
```

```
(RPAQ? *PRINT-SEMICOLON-COMMENTS* NIL)

(RPAQ? *BACKQUOTE-WRAPPERS* '(BQUOTE %, %,@ %,.))

;; Prettyprintmacros for common lisp and other poor things

(DEFINEQ

(CODEWRAPPER.PRETTYPRINT
  [LAMBDA (FORM)                                          ; Edited 30-Mar-88 11:44 by bvm
    ;; Prettyprints things that wrap code like PROGN.  We usually want them to start the code on the next line, rather than put the first expression way
    ;; to the right of all the rest.
    (PRIN1 "(")
    (LET ((HERE (INDENTATION.FROM.HERE)))
         (PRIN2 (pop FORM))                               ; Print the "function" itself
         (if (NLISTP FORM)
             then                                         ; Ignore degenerate cases
                  (PRINTDEF FORM T T T FNSLST)
           else (SEQUENTIAL.PRETTYPRINT FORM HERE))
         (PRIN1 ")")
         NIL])

(PROG1.PRETTYPRINT
  [LAMBDA (EXPR)                                          ; Edited 14-Apr-88 18:39 by bvm
    ;; Prettyprinter advice for PROG1, CL:IF, UNLESS, etc.  Default way's main problem is that if the first expression is a non-list but some later
    ;; expression is a list, it doesn't put ALL the subsequent expressions equally indented.  Thus, you get something like (PROG1 A (expression) <cr>
    ;; (expression) ...)
    (if [OR [NLISTP (CDR (LISTP (CDR EXPR]
            (AND (NLISTP (CDDDR EXPR))
                 (for E in (LISTP (CADDR EXPR)) never (LISTP E]
        then                                             ; 2 or fewer elements, or 3 elements, the last of which is very
                                                         ; simple--let default prettyprinter do it
             EXPR
      else (PRIN1 "(")
           (LET [(HERE (INDENTATION.FROM.HERE))
                 (LEFT (PROGN (PRIN2 (pop EXPR))         ; Print the car of form
                              (SPACES 1)
                              (DSPXPOSITION]
                (DECLARE (SPECVARS LEFT))
                (if (OR (if (>= HERE LEFT)
                            then                         ; Default indentation wants to be greater than the function length,
                                                         ; so change it to here
                                 (SETQ HERE LEFT))
                        (NLISTP (CAR EXPR))
                        (FITP (CAR EXPR)))
                    then (SUPERPRINT (CAR EXPR)
                                     EXPR NIL *STANDARD-OUTPUT*)   ; Print the first element right at this position
                         (pop EXPR))
                (SEQUENTIAL.PRETTYPRINT EXPR HERE))
           (PRIN1 ")")                                   ; Return NIL to say we handled it
           NIL])

(CASE.PRETTYPRINT
  [LAMBDA (EXPR)                                         ; Edited 26-Apr-88 10:52 by bvm
    (if (NLISTP (CDR EXPR))
        then                                             ; Degenerate case--punt
             EXPR
      else
      (PRIN1 "(")
      (LET
       ((HERE (INDENTATION.FROM.HERE))
        (LEFT (PROGN (PRIN2 (pop EXPR))                  ; Print the car of form
                     (SPACES 1)
                     (DSPXPOSITION)))
        (TAIL EXPR)
        INNERLEFT CASE)
       (DECLARE (SPECVARS LEFT TAIL))
       (if (OR (if (>= HERE LEFT)
                   then                                  ; Default indentation wants to be greater than the function length,
                                                         ; so change it to here
                        (SETQ HERE LEFT))
               (NLISTP (CAR TAIL))
               (FITP (CAR TAIL)))
           then (SUPERPRINT (CAR TAIL)
                            TAIL NIL *STANDARD-OUTPUT*)  ; Print the first element right at this position
                (pop TAIL))
       [SETQ INNERLEFT (+ (SETQ LEFT HERE)
                          (TIMES 3 (CHARWIDTH (CHARCODE X)
                                              *STANDARD-OUTPUT*]
       (do
        (if (NLISTP TAIL)
            then (if TAIL
                     then                                ; dotted tail?
```

```
                    (PRINENDLINE LEFT *STANDARD-OUTPUT*)
                    (PRINTDEF TAIL T T T))
              (PRIN1 ")")
              (RETURN NIL)
        elseif (SEMI-COLON-COMMENT-P (LISTP (CAR TAIL)))
          then                                          ; Print any comments stuck in between elements
              (SUPERPRINT/COMMENT (CAR TAIL)
                     *STANDARD-OUTPUT*)
              (pop TAIL)
        else                                            ; Start new line, after printing any comments
        (PRINENDLINE LEFT *STANDARD-OUTPUT*)
        (if (NLISTP (SETQ CASE (CAR TAIL)))
            then                                        ; Degenerate case?
                 (PRIN2 CASE)
          elseif (FMEMB (CAR CASE)
                        *BACKQUOTE-WRAPPERS*)
            then                                        ; backquoted case
                 (SUPERPRINT CASE TAIL NIL *STANDARD-OUTPUT*)
          else (PRIN1 "(")
               (LET (FORMFLG)
                    (DECLARE (SPECVARS FORMFLG))        ; Print the key(s) as data
                    (SUPERPRINT (CAR CASE)
                           CASE NIL *STANDARD-OUTPUT*)
                    (SPACES 1))
               [if (NLISTP (SETQ CASE (CDR CASE)))
                   then                                 ; No tail, but handle degenerates
                        (PRINTDEF CASE T T T)
                 else (SEQUENTIAL.PRETTYPRINT
                       CASE
                       (LET ((HERE (DSPXPOSITION)))
                            (if [OR (<= HERE INNERLEFT)
                                    (AND (NULL (CDR CASE))
                                         (if (LISTP (CDR CASE))
                                             then       ; Multiple things to print
                                                  NIL
                                           elseif (NLISTP (CAR CASE))
                                             then       ; Print simple consequent if space
                                                  (< (STRINGWIDTH (CAR CASE)
                                                                  *STANDARD-OUTPUT* T)
                                                     (- (DSPRIGHTMARGIN)
                                                        HERE))
                                             else (FITP CASE T]
                                then                    ; Key didn't go too far over, so just prettyprint from here
                                     HERE
                                else INNERLEFT]
               (PRIN1 ")"))
        (pop TAIL])
```

(PROGV.PRETTYPRINT

```
[LAMBDA (EXPR)                                          ; Edited 14-Apr-88 18:37 by bvm
```

;; Prettyprinter advice for PROGV. Default way's main problem is that if the vars and values are non-lists the "body" of the form doesn't get
;; uniformly indented. Thus, you get something like (PROGV vars values (expression) <cr> (expression) ...)

```
(if [OR (NLISTP (CDR EXPR))
        (LISTP (CADR EXPR))
        (NLISTP (CDR (LISTP (CDDR EXPR]
    then                                                ; 3 or fewer elements, or the second is a list--default prettyprinter
                                                        ; will do fine
         EXPR
  else (PRIN1 "(")
       (LET [(HERE (INDENTATION.FROM.HERE))
             (LEFT (PROGN (PRIN2 (pop EXPR))            ; Print the car of form
                          (SPACES 1)
                          (DSPXPOSITION]
            (DECLARE (SPECVARS LEFT))
            (SUPERPRINT (CAR EXPR)
                   EXPR NIL *STANDARD-OUTPUT*)          ; Print the first element (vars) at this position
            (pop EXPR)
            (if (OR (NLISTP (CAR EXPR))
                    (FITP (CAR EXPR)))
                then (SPACES 1)                         ; Room for next element (values) here
                     (SUPERPRINT (CAR EXPR)
                            EXPR NIL *STANDARD-OUTPUT*)
                     (pop EXPR))                        ; Finally, print the body
            (SEQUENTIAL.PRETTYPRINT EXPR HERE))
       (PRIN1 ")")                                      ; Return NIL to say we handled it
       NIL])
```

(DO.PRETTYPRINT

```
[LAMBDA (EXPR)                                          ; Edited 26-Apr-88 11:30 by bvm
```

;; Prettyprinter advice for DO, DO*, DOLIST, DOTIMES. Default way's main problem is that the body is indented at the same level as the clauses.
;; Syntax: (do clauses exit . body)

```
(if [NOT (LISTP (CDR (LISTP (CDR EXPR]
    then                                                ; 2 or fewer elements--default prettyprinter will do fine
```

```
            EXPR
      else (PRIN1 "(")
           [LET* ((START (DSPXPOSITION))
                  (HEAD (CAR EXPR))
                  (LEFT (PROGN (PRIN2 HEAD)                    ; Print the car of form
                               (SPACES 1)
                               (DSPXPOSITION)))
                  (TAIL (CDR EXPR)))
                 (DECLARE (SPECVARS LEFT TAIL))
                 (SUPERPRINT (CAR TAIL)
                        TAIL NIL *STANDARD-OUTPUT*)            ; Print the first element (var clauses) at this position
                 (pop TAIL)
                 (SELECTQ HEAD
                      ((CL:DO CL:DO*)                          ; There's another clause here
                           (SUBPRINT/ENDLINE LEFT *STANDARD-OUTPUT*)
                                                               ; Indent next at same level, printing any comments first
                           (if (LISTP TAIL)
                               then                            ; Unless degenerate case, print the second element (end test) at
                                                               ; this position
                                    (if (NULL (CAR TAIL))
                                        then                   ; Empty exit condition
                                             (PRIN1 "()")
                                      else (SUPERPRINT (CAR TAIL)
                                                       TAIL NIL *STANDARD-OUTPUT*))
                                    (pop TAIL)))
                      NIL)
                 ;; Finally, print the body, with left margin halfway between left edge of form and the initial clauses
                 (SEQUENTIAL.PRETTYPRINT TAIL (+ START (MIN (TIMES 3 SPACEWIDTH)
                                                            (IQUOTIENT (- LEFT START)
                                                                       2]
           (PRIN1 ")")                                         ; Return NIL to say we handled it
           NIL])
```

(INDENTATION.FROM.HERE

```
  [LAMBDA NIL                                                  ; Edited 28-Mar-88 18:17 by bvm
    ;; Returns X-pos about 3 chars over, for use in indenting code
    (+ (DSPXPOSITION)
       (TIMES 3 (CHARWIDTH (CHARCODE X)
                           *STANDARD-OUTPUT*])
```

(SEQUENTIAL.PRETTYPRINT

```
  [LAMBDA (TAIL LEFT)                                          ; Edited 26-Apr-88 11:15 by bvm
    (DECLARE (SPECVARS TAIL LEFT))
    ;; Print each element of tail indented at position LEFT.
    (PROG (TEM)
          (if (<= (DSPXPOSITION)
                  LEFT)
              then                                             ; Don't start with newline if we aren't to the right of the left margin
                   (GO MIDDLE))
     TOP  (if (OR (NULL TAIL)
                  (PROGN (SUBPRINT/ENDLINE LEFT *STANDARD-OUTPUT*)
                         (NULL TAIL)))
              then                                             ; Done
                   (RETURN))
     MIDDLE
          (if (NLISTP TAIL)
              then                                             ; Degenerate tail
                   (RETURN (PRINTDEF TAIL T T T))
            elseif (AND (LISTP (CDR TAIL))
                        (FMEMB (CAR TAIL)
                               *BACKQUOTE-WRAPPERS*)
                        (NULL (CDDR TAIL))
                        (SETQ TEM (GET (CAR TAIL)
                                       'PRETTYWRAPPER))
                        (SETQ TEM (CL:FUNCALL TEM TAIL *STANDARD-OUTPUT*)))
              then                                             ; Dotted backquote tail (sigh)
                   (RETURN (SUBPRINT/WRAPPERTAIL TAIL TEM)))
          (SUPERPRINT (CAR TAIL)
                 TAIL NIL *STANDARD-OUTPUT*)
          (pop TAIL)
          (GO TOP])

)

(ADDTOVAR PRETTYPRINTMACROS
          (UNINTERRUPTABLY . CODEWRAPPER.PRETTYPRINT)
          (CL:UNWIND-PROTECT . CODEWRAPPER.PRETTYPRINT)
          (RESETLST . CODEWRAPPER.PRETTYPRINT)
          (CL:BLOCK . PROG1.PRETTYPRINT)
          (CL:IF . PROG1.PRETTYPRINT)
          (PROG1 . PROG1.PRETTYPRINT)
          (CL:WHEN . PROG1.PRETTYPRINT)
```

```
          (CL:UNLESS . PROG1.PRETTYPRINT)
          (WITH-READER-ENVIRONMENT . PROG1.PRETTYPRINT)
          (CL:CATCH . PROG1.PRETTYPRINT)
          (CASE . CASE.PRETTYPRINT)
          (CL:ECASE . CASE.PRETTYPRINT)
          (CL:ETYPECASE . CASE.PRETTYPRINT)
          (CL:TYPECASE . CASE.PRETTYPRINT)
          (CL:PROGV . PROGV.PRETTYPRINT)
          (WITH.MONITOR . PROG1.PRETTYPRINT)
          (CL:DO* . DO.PRETTYPRINT)
          (CL:DO . DO.PRETTYPRINT)
          (CL:DOLIST . DO.PRETTYPRINT)
          (CL:DOTIMES . DO.PRETTYPRINT))

(ADDTOVAR PRETTYEQUIVLST (PROG* . PROG)
                         (CL:COMPILER-LET . LET))

(DECLARE%: EVAL@COMPILE DOCOPY

(CL:PROCLAIM '(CL:SPECIAL DEFAULTFONT BOLDFONT USERFONT SYSTEMFONT CLISPFONT BIGFONT PRETTYPRINTMACROS))
)

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS CLISPARRAY CHANGESARRAY AVERAGEFNLENGTH %#CAREFULCOLUMNS AVERAGEVARLENGTH FONTWORDS FONTFNS
       CLISPCHARS FUNNYATOMLST PRETTYEQUIVLST COMMENTFLG *BACKQUOTE-WRAPPERS*)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: DSPRETTY PRINTDEF SUPERPRINT SUPERPRINT0 SUBPRINT SUBPRINT1 PRINTPROG PRINTPROGVARS PRINTSQ BACKARROWP
       RPARS FITP DSFIT1 DSFIT2 (ENTRIES PRINTDEF SUPERPRINT FITP)
       (SPECVARS TAIL LEFT FNSLST FIRSTPOS COMMENTCOL FORMFLG FILEFLG)
       (LOCALFREEVARS TAILFLG CHANGEFLG))
)

(DECLARE%: DONTEVAL@LOAD

(FILESLOAD (LOADCOMP)
       DSPRINTDEF)
)
)

(PUTPROPS NEWPRINTDEF COPYRIGHT ("Venue" 1982 1983 1984 1985 1986 1987 1988 1990 1991 1993))
```

FUNCTION INDEX

VARIABLE INDEX