

File created: 9-Jun-2021 22:50:15 {DSK}<Users>kaplan>Local>medley3.5>git-medley>sources>APRINT.;11

changes to: (VARS APRINTCOMS)

previous date: 10-May-2021 15:46:22 {DSK}<Users>kaplan>Local>medley3.5>git-medley>sources>APRINT.;10

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1982-1988, 1990-1991, 2020-2021 by Venue & Xerox Corporation.

(RPAQQ **APRINTCOMS**

```
[ (COMS                                     ; User-level print functions
  (FNS PRIN1 PRIN2 PRIN3 PRIN4 PRINT PRINTCCODE \PRINTCCODE PRINTLEVEL RADIX SPACES TERPRI FRESHLINE
    DEFPRINT LINELENGTH))
  (INITVARS (PLVLFILFLG NIL)
    (\LINELENGTH 82)
    (\FLOATFORMAT T)
    (PRXFLG NIL)
    (*PRINT-BASE* 10)
    (*READ-BASE* 10)
    (*PRINT-RADIX* NIL)
    (*PRINT-ESCAPE* T)
    (*PRINT-CASE* ' :UPCASE)
    (*PRINT-GENSYM* T)
    (*PRINT-LEVEL* NIL)
    (*PRINT-LENGTH* NIL)
    (*PRINT-PRETTY* NIL)
    (*PRINT-CIRCLE* NIL)
    (*PRINT-ARRAY* NIL)
    (*PRINT-CIRCLE-HASHTABLE* NIL)
    (*PACKAGE* NIL)
    (*KEYWORD-PACKAGE* NIL)
    (*INTERLISP-PRIN1-CASE* ' :UPCASE)
    (\DEFPRINTFNS NIL))

  (COMS                                     ; PRINT internals
  (FNS PRINT-CIRCLE-LOOKUP PRINT-CIRCLE-LABEL-P PRINT-CIRCLE-SCAN PRINT-CIRCLE-ENTER)
  (FNS \PRINDATUM \PRINT-USING-DEFPRINT \PRINT-USING-ADDRESS \ELIDE.PRINT.ELEMENT
    \ELIDE.ELEMENT.CHAR \ELIDE.PRINT.TAIL \ELIDE.TAIL.STRING \CKPOSBOUT \CKPOSSOUT \CONVERTNUMBER
    \LITPRIN \LITPRIN.INTERNAL \SYMBOL.ESCAPE.COUNT \NUMERIC.PNAMEP \PRINSTACKP \PRINTADDR
    \PRINSTRING \SOUT)
  (DECLARE%: EVAL@COMPILE DONTCOPY (MACROS .FILELINELENGTH.)
    (FUNCTIONS \PRINDATUM-LISTP)
    (EXPORT (MACROS .SPACECHECK. \CHECKRADIX)))
  (FNS \INVALID.RADIX)
  (SPECVARS \THISFILELINELENGTH))

  (COMS                                     ; Internal printing
  (FNS \MAPNAME \MAPNAME.INTERNAL PNAMESTREAMP)
  (DECLARE%: DONTCOPY (RESOURCES \MAPNAMESTREAM)
    (MACROS PNAMESTREAMP))
  (INITRESOURCES \MAPNAMESTREAM)
  [INITVARS (\PNAMEDEVICE (NCREATE 'FDEV (\GETDEVICEFROMHOSTNAME 'NULL T)
    (GLOBALVARS \PNAMEDEVICE))

  (COMS                                     ; Obsolete
  (FNS \MAPCHARS))
  (DECLARE%: EVAL@COMPILE DOCOPY (ADDVARS (SYSSPECVARS *PRINT-BASE* *READ-BASE* *PRINT-RADIX*
    *PRINT-ESCAPE* *PRINT-CASE* *PRINT-GENSYM* *PRINT-LEVEL*
    *PRINT-LENGTH* *PRINT-PRETTY* *PRINT-CIRCLE*
    *PRINT-ARRAY* *PACKAGE*))

  (COMS                                     ; PRINTNUM and friends
  (FNS PRINTNUM FLTGMT \CHECKFLTGMT PRINTNUM-TO-STRING)
  (MACROS NUMFORMATCODE)
  (INITVARS (NILNUMPRINTFLG)))
  (LOCALVARS . T)
  (GLOBALVARS \LINELENGTH \FLOATFORMAT PRXFLG \DEFPRINTFNS)
  (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
    (NLAML)
    (LAMA]))
```

:: User-level print functions

(DEFINEQ

(**PRIN1**

[LAMBDA (X FILE)

(* bvm%: "29-Sep-86 23:59")

::: Like PRIN2 but no escaping. Also implies no radix qualifiers, although Common Lisp separates *PRINT-RADIX* from *PRINT-ESCAPE* -- might
::: want to bind *PRINT-RADIX* to (AND (fetch (READTABLEP COMMONLISP) of *READTABLE*) *PRINT-RADIX*)

```
(LET* [(STRM (\GETSTREAM FILE 'OUTPUT))
  (OBEY-PRINT-LEVEL (OR (ffetch (READTABLEP COMMONLISP) of (\DTEST *READTABLE* 'READTABLE))
    (OR (\OUTTERMP STRM)
```

```

PLVLFLEFLG]
(LET ((*PRINT-ESCAPE* NIL)
      (*PRINT-RADIX* NIL)
      (*PRINT-LEVEL* (AND OBEY-PRINT-LEVEL *PRINT-LEVEL*))
      (*PRINT-LENGTH* (AND OBEY-PRINT-LEVEL *PRINT-LENGTH*))
      (*PRINT-CASE* (OR *INTERLISP-PRIN1-CASE* *PRINT-CASE*))
      \THISFILELINELENGTH)
  (DECLARE (SPECVARS *PRINT-RADIX* *PRINT-ESCAPE* *PRINT-LEVEL* *PRINT-LENGTH* *PRINT-CASE*
                  \THISFILELINELENGTH)
           ; *PRINT-CASE* because too many things in Interlisp print
           ; things expecting the symbol's pname
  (SETQ \THISFILELINELENGTH (.FILELINELENGTH. STRM))
  (\PRINDATUM X STRM 0)
  X])

```

(PRIN2

```

[LAMBDA (X FILE RDTBL) ; Edited 20-Jan-87 17:04 by bvm:
  (LET* [(STRM (\GETSTREAM FILE 'OUTPUT))
        (OBEY-PRINT-LEVEL (OR (fetch (READTABLEP COMMONLISP) of (SETQ RDTBL (\GTREADTABLE RDTBL)))
                              (OR (\OUTERMP STRM)
                                  PLVLFLEFLG]
        (LET ((*READTABLE* RDTBL)
              (*PRINT-ESCAPE* T)
              (*PRINT-LEVEL* (AND OBEY-PRINT-LEVEL *PRINT-LEVEL*))
              (*PRINT-LENGTH* (AND OBEY-PRINT-LEVEL *PRINT-LENGTH*))
              (*PACKAGE* (if (fetch (READTABLEP USESILPACKAGE) of RDTBL)
                             then *INTERLISP-PACKAGE*
                             else *PACKAGE*))
              \THISFILELINELENGTH)
        (DECLARE (SPECVARS *PRINT-ESCAPE* *READTABLE* *PRINT-LEVEL* *PRINT-LENGTH* *PACKAGE*
                          \THISFILELINELENGTH))
        (SETQ \THISFILELINELENGTH (.FILELINELENGTH. STRM))
        (\PRINDATUM X STRM 0)
        X])

```

(PRIN3

```

[LAMBDA (X FILE) (* bvm%: "29-Sep-86 23:59")

```

;;; Like PRIN1 but no linelength checking

```

(LET* [(STRM (\GETSTREAM FILE 'OUTPUT))
      (OBEY-PRINT-LEVEL (OR (ffetch (READTABLEP COMMONLISP) of (\DTEST *READTABLE* 'READTABLEP))
                            (OR (\OUTERMP STRM)
                                PLVLFLEFLG]
      (LET ((*PRINT-ESCAPE* NIL)
            (*PRINT-RADIX* NIL)
            (*PRINT-LEVEL* (AND OBEY-PRINT-LEVEL *PRINT-LEVEL*))
            (*PRINT-LENGTH* (AND OBEY-PRINT-LEVEL *PRINT-LENGTH*))
            (*PRINT-CASE* (OR *INTERLISP-PRIN1-CASE* *PRINT-CASE*))
            \THISFILELINELENGTH)
        (DECLARE (SPECVARS *PRINT-RADIX* *PRINT-ESCAPE* *PRINT-LEVEL* *PRINT-LENGTH* \THISFILELINELENGTH
                          ))
        (\PRINDATUM X STRM 0)
        X])

```

(PRIN4

```

[LAMBDA (X FILE RDTBL) ; Edited 20-Jan-87 17:05 by bvm:

```

;;; Like PRIN2 but doesn't check linelength

```

(LET* [(STRM (\GETSTREAM FILE 'OUTPUT))
      (OBEY-PRINT-LEVEL (OR (fetch (READTABLEP COMMONLISP) of (SETQ RDTBL (\GTREADTABLE RDTBL)))
                            (OR (\OUTERMP STRM)
                                PLVLFLEFLG]
      (LET ((*READTABLE* RDTBL)
            (*PRINT-ESCAPE* T)
            (*PRINT-LEVEL* (AND OBEY-PRINT-LEVEL *PRINT-LEVEL*))
            (*PRINT-LENGTH* (AND OBEY-PRINT-LEVEL *PRINT-LENGTH*))
            (*PACKAGE* (if (fetch (READTABLEP USESILPACKAGE) of RDTBL)
                           then *INTERLISP-PACKAGE*
                           else *PACKAGE*))
            \THISFILELINELENGTH)
        (DECLARE (SPECVARS *PRINT-ESCAPE* *READTABLE* *PRINT-LEVEL* *PRINT-LENGTH* *PACKAGE*
                          \THISFILELINELENGTH))
        (\PRINDATUM X STRM 0)
        X])

```

(PRINT

```

[LAMBDA (X FILE RDTBL) (* bvm%: " 9-May-86 23:08")

```

```

(LET [(STRM (\GETSTREAM FILE 'OUTPUT))
      (PRIN2 X STRM RDTBL)
      (\OUTCHAR STRM (CHARCODE EOL))
      X])

```

(PRINTCCODE

```
[LAMBDA (CHARCODE FILE)
  (\OUTCHAR (\GETSTREAM FILE 'OUTPUT)
    (COND
      ((\CHARCODEP CHARCODE)
        CHARCODE)
      (T (\ILLEGAL.ARG CHARCODE]))
```

(* bvm%: " 9-May-86 22:44")

(\PRINTCCODE

```
[LAMBDA (CHARCODE FILE)
  ;; A copy of PRINTCCODE to be used in internal functions where \OUTCHAR doesn't have to be compiled open but which we don't want to screw
  ;; up if the user breaks PRINTCCODE
  (\OUTCHAR (\GETSTREAM FILE 'OUTPUT)
    (COND
      ((\CHARCODEP CHARCODE)
        CHARCODE)
      (T (\ILLEGAL.ARG CHARCODE]))
```

; Edited 5-May-2021 09:56 by rmk:

(PRINTLEVEL

```
[LAMBDA (CARVAL CDRVAL)
```

(* bvm%: " 9-May-86 22:47")

;;; Sets Interlisp print level to the given values in CAR and CDR directions. These correspond to *PRINT-LEVEL* and *PRINT-LENGTH* in Common Lisp

```
[COND
  ((LISTP CARVAL)
   (SETQ CDRVAL (CDR CARVAL))
   (SETQ CARVAL (CAR CARVAL))
  (PROG1 (CONS (OR *PRINT-LEVEL* -1)
               (OR *PRINT-LENGTH* -1)))
  [COND
    (CARVAL (SETQ *PRINT-LEVEL* (AND (IGEQ CARVAL 0)
                                      CARVAL))
  [COND
    (CDRVAL (SETQ *PRINT-LENGTH* (AND (IGEQ CDRVAL 0)
                                       CDRVAL)))]])
```

(RADIX

```
[LAMBDA (N)
  (PROG1 *PRINT-BASE*
    (AND N (SETQ *PRINT-BASE* (\CHECKRADIX N))))]
```

(* bvm%: " 5-May-86 10:56")

(SPACES

```
[LAMBDA (N FILE)
  [PROG ((STREAM (\GETSTREAM FILE 'OUTPUT))
        \THISFILELINELENGTH)
    (SETQ \THISFILELINELENGTH (.FILELINELENGTH. STREAM))
    (.SPACECHECK. STREAM N)
    (FRPTQ N (\OUTCHAR STREAM (CHARCODE SPACE)
  NIL])
```

(* rmk%: "21-OCT-83 12:32")

(TERPRI

```
[LAMBDA (FILE)
  (\OUTCHAR (\GETSTREAM FILE 'OUTPUT)
    (CHARCODE EOL))
  NIL])
```

(* rmk%: "21-OCT-83 12:31")

(FRESHLINE

```
[LAMBDA (STREAM)
  ;; Adjusts the STREAM to be at a new line -- does equivalent of TERPRI unless it is already 'sitting at the beginning of a line'
  (COND
    ([NEQ 0 (fetch CHARPOSITION of (COND
      ((AND (type? STREAM STREAM)
            (WRITEABLE STREAM))
        STREAM)
      (T (SETQ STREAM (GETSTREAM STREAM 'OUTPUT))
        (\OUTCHAR STREAM (CHARCODE EOL))
      T])
```

(* rmk%: "22-AUG-83 13:48")

(DEFPRINT

```
[LAMBDA (TYPE FN)
  (AND (FIXP TYPE)
    (SETQ TYPE (\TYPENAMEFROMNUMBER TYPE)))
  (PROG ((F (FASSOC TYPE \DEFPRINTFNS))
    [COND
      (F (SETQ \DEFPRINTFNS (DREMOVE F \DEFPRINTFNS))
    [COND
```

(* rmk%: "28-APR-80 12:04")

; The FIXP case should never occur

```
(FN (SETQ \DEFPRINTFNS (CONS (CONS TYPE FN)
                             \DEFPRINTFNS]
(RETURN (CDR F]))
```

(LINELENGTH

```
[LAMBDA (N FILE)
```

(* bvm%: "11-Mar-86 14:56")

:: Sets to N the linelength of FILE -- defaults to primary output file

```
(LET [(STREAM (\GETSTREAM FILE 'OUTPUT)
              (PROG1 (fetch (STREAM LINELENGTH) of STREAM)
                    [AND N (COND
                        ((AND (NUMBERP N)
                             (ILESSP N 1))
                         (\ILLEGAL.ARG N))
                        (T (replace (STREAM LINELENGTH) of STREAM with (COND
                                ((EQ N T)
                                 ; Infinite
                                 MAX.SMALLP)
                                (T (FIX N))))])])])
```

)

- (RPAQ? PLVLFILEFLG NIL)
- (RPAQ? \LINELENGTH 82)
- (RPAQ? \FLOATFORMAT T)
- (RPAQ? PRXFLG NIL)
- (RPAQ? *PRINT-BASE* 10)
- (RPAQ? *READ-BASE* 10)
- (RPAQ? *PRINT-RADIX* NIL)
- (RPAQ? *PRINT-ESCAPE* T)
- (RPAQ? *PRINT-CASE* ' :UPCASE)
- (RPAQ? *PRINT-GENSYM* T)
- (RPAQ? *PRINT-LEVEL* NIL)
- (RPAQ? *PRINT-LENGTH* NIL)
- (RPAQ? *PRINT-PRETTY* NIL)
- (RPAQ? *PRINT-CIRCLE* NIL)
- (RPAQ? *PRINT-ARRAY* NIL)
- (RPAQ? *PRINT-CIRCLE-HASHTABLE* NIL)
- (RPAQ? *PACKAGE* NIL)
- (RPAQ? *KEYWORD-PACKAGE* NIL)
- (RPAQ? *INTERLISP-PRIN1-CASE* ' :UPCASE)
- (RPAQ? \DEFPRINTFNS NIL)

:: PRINT internals

(DEFINEQ

(PRINT-CIRCLE-LOOKUP

```
[LAMBDA (OBJECT) (* Pavel "16-Oct-86 21:13")
  (LET ((TABLEENTRY (GETHASH OBJECT *PRINT-CIRCLE-HASHTABLE*))
        (CASE TABLEENTRY
          ((T1 NIL) (CL:VALUES NIL NIL))
          (T2 (CL:VALUES (PROG1 (CONCAT (CHARACTER (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*)
                                             *PRINT-CIRCLE-NUMBER* "=")
                                     (CL:SETF (CL:GETHASH OBJECT *PRINT-CIRCLE-HASHTABLE*)
                                             *PRINT-CIRCLE-NUMBER*)
                                     (CL:INCF *PRINT-CIRCLE-NUMBER*))
                        T))
          (CL:OTHERWISE (CL:IF (NUMBERP TABLEENTRY)
                               (CL:VALUES (CONCAT (CHARACTER (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*)
                                                                    TABLEENTRY "#")
                                                  NIL)
                                           (CL:ERROR "Print-circle-lookup hashtable error!"))))))))
```

(PRINT-CIRCLE-LABEL-P

```
[CL:LAMBDA (OBJECT) ; (* jrb%: "30-Jun-86 23:04")
  (DECLARE (CL:SPECIAL *PRINT-CIRCLE-HASHTABLE*))
  (CL:BLOCK PRINT-CIRCLE-LABEL-P
    (LET ((TABLEENTRY (GETHASH OBJECT *PRINT-CIRCLE-HASHTABLE*)))
      (COND
        ((EQ TABLEENTRY 'T2))
        ((CL:INTEGERP TABLEENTRY)
         TABLEENTRY)
        (T NIL))))])
```

(PRINT-CIRCLE-SCAN

; Edited 16-Jan-87 15:53 by jrb:

```
[CL:LAMBDA (OBJECT)
  (DECLARE (CL:SPECIAL *PRINT-ARRAY*))
  (CL:TYPECASE OBJECT
    (CONS [COND
      ((NOT (PRINT-CIRCLE-ENTER OBJECT))
       (PRINT-CIRCLE-SCAN (CAR OBJECT)))
      (PRINT-CIRCLE-SCAN (CDR OBJECT))]
      (CL::STRUCTURE-OBJECT [COND
        ((AND XCL:*PRINT-STRUCTURE* (NOT (PRINT-CIRCLE-ENTER OBJECT)))
         (CL:MAPCAR [FUNCTION (LAMBDA (DESCRIPTOR)
           (PRINT-CIRCLE-SCAN (FETCHFIELD DESCRIPTOR OBJECT]
            (CL::STRUCTURE-POINTER-SLOTS (CL:TYPE-OF OBJECT]))
          (CL:ARRAY T) [COND
            ((AND *PRINT-ARRAY* (NOT (PRINT-CIRCLE-ENTER OBJECT)))
             ; No need to walk array if we're not printing them
              (LET* [(ASIZE (CL:ARRAY-TOTAL-SIZE OBJECT))
                (VARRAY (COND
                  ((> (CL:ARRAY-RANK OBJECT)
                     1)
                   (CL:MAKE-ARRAY ASIZE :DISPLACED-TO OBJECT))]
                  (T OBJECT]
                (CL:DOTIMES (X ASIZE)
                 (PRINT-CIRCLE-SCAN (CL:AREF VARRAY X)))))]))])
```

(PRINT-CIRCLE-ENTER

; Edited 31-Mar-87 19:16 by jrb:

```
[CL:LAMBDA (OBJECT)
  (DECLARE (CL:SPECIAL *PRINT-CIRCLE-HASHTABLE* THERE-ARE-CIRCLES))
  (CASE (CL:GETHASH OBJECT *PRINT-CIRCLE-HASHTABLE*)
    (NIL)
    (CL:SETF (CL:GETHASH OBJECT *PRINT-CIRCLE-HASHTABLE*)
              'T1)
    NIL)
    (T1
     (CL:SETF (CL:GETHASH OBJECT *PRINT-CIRCLE-HASHTABLE*)
               'T2)
     (SETQ THERE-ARE-CIRCLES T)
     T)
    (T2 T)
    (CL:OTHERWISE (CL:ERROR "Print-circle-enter hashtable error!")))]])
```

(DEFINEQ

(\PRINDATUM

; Edited 11-Feb-91 14:34 by jds

```
[LAMBDA (OBJECT STREAM CPL)
  (DECLARE (USEDFFREE *READTABLE* *PRINT-RADIX* *PRINT-BASE* *PRINT-ESCAPE*))
  (SELECTC (NTYPX OBJECT)
    ((LIST \LITATOM \NEW-ATOM)
     (\LITPRIN OBJECT STREAM))
    (\LISTP ;; macro call that uses the arguments already bound, to save a fn call.
     (\PRINDATUM-LISTP))
    ((LIST \SMALLP \FIXP)
     (WITH-RESOURCES (\NUMSTR \NUMSTR1)
      (\CKPOSSOUT STREAM (\CONVERTNUMBER OBJECT (\CHECKRADIX *PRINT-BASE*)
        T
        (AND (if (fetch (READTABLEP COMMONLISP) of *READTABLE*)
                 then ; Common Lisp controlled solely by this var
                  *PRINT-RADIX*
                 else ; Interlisp prints radix if it is not 10 and we are prin2
                  (AND *PRINT-ESCAPE* (NEQ *PRINT-BASE* 10)))
                *READTABLE*))
              \NUMSTR \NUMSTR1))))
    (\FLOATP [WITH-RESOURCES (\NUMSTR \NUMSTR1)
              (\CKPOSSOUT STREAM (\CONVERT.FLOATING.NUMBER OBJECT \NUMSTR \NUMSTR1
                (COND
                  ((AND (PNAMESTREAMP STREAM)
                       (NOT PRXFLG))
                   ;; The pname of a number is unaffected by RADIX unless PRXFLG is true. This seems
                   ;; silly, but assorted code will break otherwise
                   T)
                T)
              T)
    T)
  T)
```

```

(T \FLOATFORMAT])
(\STACKP (\PRINSTACKP OBJECT STREAM)
(COND
  ((STRINGP OBJECT)
  (\PRINSTRING OBJECT STREAM))
  ((TYPENAMEP OBJECT 'CL::STRUCTURE-OBJECT)
  ;; this is a structure, don't use defprint.
  (CL::PRINT-STRUCTURE-INSTANCE OBJECT STREAM CPL))
  ((TYPENAMEP OBJECT 'T)
  ;; this is a common-loops object, since it is a sub-class of t, so call the print-instance method.
  (PRINT-INSTANCE OBJECT STREAM 0))
  (T (\PRINT-USING-DEFPRINT OBJECT STREAM CPL])

```

(\PRINT-USING-DEFPRINT

; Edited 18-Dec-86 12:22 by bvm:

```

[LAMBDA (X STREAM CPL)
  (DECLARE (USEDFREE *PRINT-LEVEL*))
  (LET* ((TYPE (TYPENAME X))
        (FN (FASSOC TYPE \DEFPRINTFNS)))
    (COND
      ([OR (NULL FN)
          (NULL (SETQ FN (LET [( *PRINT-LEVEL* (AND *PRINT-LEVEL* (DIFFERENCE *PRINT-LEVEL*
                                                                                   (OR CPL 0])
                                                                                   X STREAM 0]
          (CL:FUNCCALL (CDR FN)
                      X STREAM 0]

```

; This way recursive calls to PRINT etc will be at the 'right' level

;; No defined printer, or printer declined to do anything

```

(\PRINT-USING-ADDRESS X STREAM CPL))
(LISTP FN)

```

;; PRIN1 the CAR (usually a macro char) and PRIN2 the CDR. Nowadays there is little reason for a defprint fn to not do its own printing

```

(AND (CAR FN)
  (LET (*PRINT-ESCAPE*)
    (\PRINDATUM (CAR FN)
                STREAM)))
(AND (CDR FN)
  (\PRINDATUM (CDR FN)
              STREAM CPL])

```

(\PRINT-USING-ADDRESS

```

(CL:LAMBDA (X STREAM CPL)
  (CL:BLOCK \PRINT-USING-ADDRESS
  (LET ((TYPE (TYPENAME X)))
    (COND
      ((fetch (READTABLEP COMMONLISP) of *READTABLE*)
      (.SPACECHECK. STREAM 2)
      (\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
      (\OUTCHAR STREAM (CHARCODE "<"))
      (AND TYPE (\LITPRIN TYPE STREAM))
      (\CKPOSSOUT STREAM " @ ")
      (\PRINTADDR X STREAM)
      (\CKPOSBOUT STREAM (CHARCODE ">")))
      (T (\CKPOSBOUT STREAM (CHARCODE {}))
        (AND TYPE (\LITPRIN TYPE STREAM))
        (\CKPOSBOUT STREAM (CHARCODE {}))
        (\OUTCHAR STREAM (CHARCODE "#"))
        (\PRINTADDR X STREAM]
    T)))

```

(\ELIDE.PRINT.ELEMENT

(* jrb%: "29-Jun-86 21:05")

```

[LAMBDA (STREAM)
  (\OUTCHAR STREAM (\ELIDE.ELEMENT.CHAR])

```

(\ELIDE.ELEMENT.CHAR

(* jrb%: "29-Jun-86 21:04")

```

[LAMBDA NIL
  (COND
    ((fetch (READTABLEP COMMONLISP) of *READTABLE*)
    (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
    (T (CHARCODE "&"))

```

(\ELIDE.PRINT.TAIL

(* jrb%: "29-Jun-86 21:06")

```

[LAMBDA (STREAM NOSPACEP)

```

;;; Prints the appropriate elision indicator for elements beyond *PRINT-DEPTH* according to the read table we're using. Prints first a space unless NOSPACEP

```

(COND
  ((NOT NOSPACEP)
  (\OUTCHAR STREAM (CHARCODE SPACE]

```



```

then ; two-digit radix
(RPLCHARCODE NS (add POS -1)
(+ (CHARCODE 0)
(IQUOTIENT R 10]
(RPLCHARCODE NS (add POS -1)
(fetch (READTABLEP HASHMACROCHAR) of RDTBL))
(SUBSTRING NS POS END NSB])

```

(\LITPRIN

```

[LAMBDA (X STREAM) ; Edited 14-Apr-87 14:49 by jrb:
(DECLARE (USEDFREE \THISFILELINELENGTH *PRINT-ESCAPE* *READTABLE* *PACKAGE* *PRINT-GENSYM* *PRINT-CASE*))
(COND
(*PRINT-ESCAPE* (LET ((RDTBL *READTABLE*)
PKG PKGSEPR)
[COND
(*PACKAGE* ; This is NIL until packages get turned on
(COND
(EQ *PACKAGE* (SETQ PKG (fetch (CL:SYMBOL PACKAGE) of X)))
; No prefix needed in current package
(SETQ PKG NIL))
[ (NULL PKG) ; Uninterned. Print something if flag is on
(COND
(*PRINT-GENSYM* ; Print #: as prefix. Not PACKAGECHAR here because colon
; hardwired into hashmacro dispatch.
(RPLCHARCODE (SETQ PKGSEPR (ALLOCSTRING 2 (CHARCODE ":")))
1
(fetch (READTABLEP HASHMACROCHAR) of RDTBL]
(EQ PKG *KEYWORD-PACKAGE*)
; Keywords get single colon, no prefix
(SETQ PKGSEPR (ALLOCSTRING 1 (fetch (READTABLEP PACKAGECHAR)
of RDTBL)))
(SETQ PKG NIL))
(FIND-EXACT-SYMBOL X *PACKAGE*)
;; Symbol is accessible in current package, either by being imported or by inheritance. This is a
;; messy test, which is why we test for special case of PKG being the current package first above. No
;; prefix needed here.
(SETQ PKG NIL))
(T ; Package qualifier is needed; we need only know now whether symbol is internal or
;; external in its home package.
(SETQ PKGSEPR (ALLOCSTRING (COND
(EQ X (FIND-EXTERNAL-SYMBOL X PKG))
; X is external in PKG, use single colon
1)
(T 2))
(fetch (READTABLEP PACKAGECHAR) of RDTBL]
(\LITPRIN.INTERNAL X RDTBL STREAM (AND PKG (PACKAGE-NAME-AS-SYMBOL PKG))
PKGSEPR \THISFILELINELENGTH)))
(T (.SPACECHECK. STREAM (\NATOMCHARS X))
;; Following code munged to match \LITPRIN.INTERNAL's handling of :CAPITALIZE
(for C inatom X bind (DOWNCASE _ (AND (fetch (READTABLEP CASEINSENSITIVE) of *READTABLE*)
(SELECTQ *PRINT-CASE*
((:DOWNCASE :CAPITALIZE)
*PRINT-CASE*)
NIL)))
(WAS-ALPHA _ NIL)
do (\OUTCHAR STREAM (if DOWNCASE
then ; may have to change case
(if (AND (>= C (CHARCODE A))
(<= C (CHARCODE Z)))
then (if (OR (NEQ DOWNCASE :CAPITALIZE)
(PROG1 WAS-ALPHA (SETQ WAS-ALPHA T)))
then ; for :capitalize, lower all but the characters that start "words",
; i.e., those immediately after a non-alphanumeric
(+ C (- (CHARCODE a)
(CHARCODE A)))
else C)
else [if (EQ DOWNCASE :CAPITALIZE)
then ; C not upper-case. Set WAS-ALPHA if it's lowercase alpha or
; numeric
(SETQ WAS-ALPHA (OR (AND (>= C (CHARCODE a))
(<= C (CHARCODE z)))
(AND (>= C (CHARCODE 0))
(<= C (CHARCODE 9)
C)
else C])

```

(\LITPRIN.INTERNAL

```

[LAMBDA (SYMBOL RDTBL STREAM PKGNAME PKGSEPR CHECKLENGTH) ; Edited 18-Dec-86 17:33 by bvm:
;;; Print SYMBOL to STREAM according to RDTBL, preceded by PKGNAME (if non-NIL) and/or PKGSEPR. PKGNAME is a symbol, PKGSEPR is a
;;; string. If CHECKLENGTH is true, need to check that there is room for printing all three parts on this line; else caller has verified that there is room

```



```

(>= C (CHARCODE A))
then (if (OR (NEQ DOWNCASE :CAPITALIZE)
             (PROG1 WAS-ALPHA (SETQ WAS-ALPHA T))))
      then
        ; for :capitalize, lower all but the characters that start "words",
        ; i.e., those immediately after a non-alphanumeric
        (+ C (- (CHARCODE a)
                (CHARCODE A)))
      else C)
else [if (EQ DOWNCASE :CAPITALIZE)
        then ; C not upper-case. It's also not lowercase, because that was
              ; caught in the CHECKESCAPE clause if any, but note if it's
              ; numeric
              (SETQ WAS-ALPHA (AND (>= C (CHARCODE 0))
                                   (<= C (CHARCODE 9))
                                C)
              else C))]
(SETQ FIRSTFLG NIL])

```

(\SYMBOL.ESCAPE.COUNT

[LAMBDA (SYMBOL RDTBL INEXACTOK)

; Edited 18-Dec-86 17:08 by bvm:

;;; Counts the number of escape characters needed to print SYMBOL by RDTBL. If RDTBL has a multiple-escape character, then we return a negative count if we're assuming it is used instead of single escapes; else a positive count. The special value -1 means the symbol is numeric, so must be quoted, but no multiple escape is available, so just escape the first character. If INEXACTOK is true and we discover we want to use multiple escape char, returns -2 immediately.

```

(for C inatom SYMBOL bind (RESULT _ 0)
                          (NESCAPES _ 0)
                          (FIRSTFLG _ T)
                          (MULTESCAPE _ (fetch (READTABLEP MULTESCAPECHAR) of RDTBL))
                          (ESCAPE _ (fetch (READTABLEP ESCAPECHAR) of RDTBL))
                          (CASEBASE _ (AND (fetch (READTABLEP CASEINSENSITIVE) of RDTBL)
                                           (fetch (ARRAYP BASE) of UPPERARRAY)))
                          (SA _ (fetch READSA of RDTBL))
                          SYN)
first (if (EQ MULTESCAPE 0)
         then ; Can't use multiple-escape
           (SETQ MULTESCAPE NIL))
do [if [OR (AND CASEBASE (ILEQ C \MAXTHINCHAR)
                (NEQ C (\GETBASEBYTE CASEBASE C)))
          (AND (fetch (READCODE ESCQUOTE) of (SETQ SYN (\SYNCODE SA C)))
               (OR FIRSTFLG (fetch (READCODE INNERESCQUOTE) of SYN]
      then ;; Need protection if char is lowercase in a case-insensitive read table or the read table says it needs it
        (add RESULT 1)
        (if MULTESCAPE
            then (if (OR (EQ C MULTESCAPE)
                        (EQ C ESCAPE))
                   then ; These have to be escaped no matter what
                     (add NESCAPES 1)
                   elseif (AND INEXACTOK (> (- RESULT NESCAPES)
                                             1))
                     then ; If at least 2 chars need escaping, better to use multiple escape,
                           ; and we can quit scanning now
                           (RETURN -2])
            else (SETQ FIRSTFLG NIL)
        finally (RETURN (if (EQ RESULT 0)
                           then ; No funny chars, check for some other perverse cases
                             (LET ((LEN (\NATOMCHARS SYMBOL)))
                               (if (EQ LEN 0)
                                   then ; The bletcherous null symbol. Shouldn't be allowed to create
                                       ; this, grumble.
                                       (if MULTESCAPE
                                           then ; Can print as ||
                                             -2
                                           else ; Single escape can't work
                                             0)
                                       elseif (AND (EQ LEN 1)
                                                  (EQ C (CHARCODE ".")))
                                           then ; Special case, dot is always escaped when by itself, and prefer
                                               ; single escape to multiple
                                               -1
                                       elseif (\NUMERIC.PNAMEP SYMBOL (if (fetch (READTABLEP COMMONLISP) of RDTBL)
                                                                              then *READ-BASE*
                                                                              else 10))
                                           then
                                           ;; Is numeric, must escape it. Note that if pname is numeric, there can't be any special chars inside it
                                           ;; needing escaping. We wait until now to test numeric on the grounds that it is more likely we will
                                           ;; print a symbol with escapable chars than one that is a potential number.
                                           (if MULTESCAPE
                                               then ; Nicer to use multiple escape around whole symbol
                                                 -2
                                               else ; Say to escape first char
                                                 -1)

```

```

                else 0))
elseif (AND MULTESCAPE (> (- RESULT NESCAPES)
                        1))
  then
    ;; The number of characters needing escaping, not counting the ones that have to be escaped in any case, is at
    ;; least two. Use two multiple-escapes and NESCAPES regular escapes for the internal escapes =
    ;; -(NESCAPES+2) total extra characters
    (- -2 NESCAPES)
  else RESULT])

```

(NUMERIC.PNAMEP

[LAMBDA (SYMBOL RADIX)

; Edited 17-Jan-2020 05:43 by rmk:
; Edited 6-Dec-91 11:27 by jds

;;; True if the chars in SYMBOL are a potential number in RADIX, which defaults to the current read base (according to current read table), OR IF the symbol consists solely of decimal points.

```

(LET* ((LASTCHARTYPE 'FIRST)
      [EFFECTIVE-RADIX (OR RADIX (COND
                                ((fetch (READTABLEP COMMONLISP) of *READTABLE*)
                                     *READ-BASE*)
                                (T 10)]
      (MAXALPHADIGIT (+ (CHARCODE A)
                       (- EFFECTIVE-RADIX 11)))
      SEENALPHADIGITS SEENDIGITS SEENDECEPT SEENEXPONENT SEENTIGHTLETTERS SEEN-ILLEGAL-SYNTAX)
  ; If RADIX is bigger than 10, this allows alphabetic digits
  (for c inpname SYMBOL do
    ;; The inpname is a nicety so it works on strings too (useful for testing) --- Note that we are assuming a
    ;; partitioning of character space as follows: (--- + / decpt) (digits) (A-Z) ( _ ^) (a-z)
    [SETQ LASTCHARTYPE
      (COND
        [(< C (CHARCODE A)) ; Numeric or funny char
         (COND
           [(< C (CHARCODE 0))
            (SELCHARQ C
                     ((- +) ; Signs anywhere but end

```

;;; RMK 2020-01-17: Anywhere at the end conforms to the Commonlisp notion of being able to extend the number syntax. But this is never going to happen here, and the consequence of not recognizing signs just at the beginning is that tokens like 1-2 will get printed as |1-2|, which is surprising. So: I'm restricting signs to first position.

```

          (CL:UNLESS (EQ LASTCHARTYPE 'FIRST)
                    (RETURN NIL))
          'SIGN)
      ("." (COND
          (SEENALPHADIGITS

```

;;; Can't have decimal point in other radices, so if we saw combinations of chars that would have been invalid in radix 10, bomb out

```

          (COND
            (SEENTIGHTLETTERS (RETURN NIL)))
            (SETQ SEENALPHADIGITS NIL))
          (SEENDECEPT

```

;;; Can't have 2 decimal points.

```

            (SETQ SEEN-ILLEGAL-SYNTAX T)))
            (SETQ MAXALPHADIGIT 0)
            (SETQ SEENDECEPT T))
          (/ (COND
             ((EQ LASTCHARTYPE 'FIRST)
              ; Can't start with ratio marker
              (RETURN NIL)))
            (RETURN NIL)))
            ((<= C (CHARCODE 9)) ; digit
             (SETQ SEENDIGITS T)
             'DIGIT)
            (T (RETURN NIL)

```

;;; Out in the wilderness.

```

            (> C (CHARCODE z))
            (RETURN NIL))
            (PROGN [COND
                    ((>= C (CHARCODE a))
                     ; Raise it
                     (SETQ C (- C (- (CHARCODE a)
                                       (CHARCODE A))

```

;;; Letter

```

                    (<= C (CHARCODE Z)))
                    ; Letter
                    ]
            [COND
              ((<= C MAXALPHADIGIT) ; Letter is a digit in this base. Can't be digit in number with
              ; decimal pt
              (COND
                (SEENDECEPT

```

;;; If there was a decimal point earlier, bail out.

```

                (RETURN NIL)))
              (SETQ SEENALPHADIGITS T)

```

```

(SELECTQ LASTCHARTYPE
  ((LETTER FIRST) ; Two letters in a row or started with letter. Notice this in case a
                  ; dec pt comes along
  (SETQ SEENTIGHTLETTERS T)
  NIL))
(T ; Potential exponent marker, only in radix 10
  (OR (IEQ 10 EFFECTIVE-RADIX)
      (RETURN NIL))
  (AND SEENEXPONENT (RETURN NIL))
  (SELECTQ LASTCHARTYPE
    ((LETTER FIRST)
     (RETURN NIL))
    (COND
      ((FMEMB C (CHARCODE (E S F D L)))
       (SETQ SEENEXPONENT T))
      (T (RETURN NIL))
      'LETTER)
    ((OR (EQ C (CHARCODE "_"))
         (EQ C (CHARCODE "^")))
     ; Extension chars, not used now but maybe some day. We're
     ; supposed to escape these
     NIL)
    (T (RETURN NIL])

```

finally

```

(RETURN (OR (AND (NOT SEEN-ILLEGAL-SYNTAX)
                 (OR SEENDIGITS SEENALPHADIGITS)
                 (NEQ LASTCHARTYPE 'SIGN))
            (AND SEENDECPT (EQ LASTCHARTYPE T)
                 (for C inpname SYMBOL always (EQ C (CHARCODE ".")))
            ; Success if there was at least one digit and didn't end in a sign.
            ; Also true if symbol consisted solely of periods.

```

(PRINSTACKP

```

[LAMBDA (X STREAM) ; (* bvm%: "11-May-86 16:09")

```

;;; Print stackp as addr/franame. If stackp is released or franame is not a symbol, print mumble

```

(.SPACECHECK. STREAM (IPLUS 1 (CONSTANT (NCHARS "<StackP ")
  (PROGN ; Longest stack address is '177,177777'
    10)
  1
  (COND
    ((RELSTKP X)
     (CONSTANT (NCHARS "released")))
    ((LITATOM (STKNAME X))
     (\NATOMCHARS (STKNAME X)))
    (T 6)
  1))
(\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
(\SOUT "<StackP " STREAM)
(\PRINTADDR X STREAM)
(\OUTCHAR STREAM (CHARCODE /))
(COND
  ((RELSTKP X)
   (\SOUT "released" STREAM))
  ((LITATOM (SETQ X (STKNAME X)))
   (\LITPRIN X STREAM))
  (T (\SOUT "*form*" STREAM)))
(\OUTCHAR STREAM (CHARCODE >])

```

(PRINTADDR

```

[LAMBDA (X STREAM) ; (* bvm%: "11-May-86 15:13")
  (WITH-RESOURCES (\NUMSTR \NUMSTR1)
    (SELECTQ (SYSTEMTYPE)
      (D (\CKPOSSOUT STREAM (\CONVERTNUMBER (\HILOC X)
        8 NIL NIL \NUMSTR \NUMSTR1))
        (\CKPOSBOUT STREAM (CHARCODE %))
        (\CKPOSSOUT STREAM (\CONVERTNUMBER (\LOLOC X)
        8 NIL NIL \NUMSTR \NUMSTR1))
      (JERICHO (\CKPOSSOUT STREAM (\CONVERTNUMBER (LOGAND \ADDRMASK (LOC X)
        8 NIL NIL \NUMSTR \NUMSTR1)))
      (VAX (\CKPOSSOUT STREAM (\CONVERTNUMBER (LOC X)
        16 T NIL \NUMSTR \NUMSTR1)))
      ((TENEX TOPS-20)
       (\CKPOSSOUT STREAM (\CONVERTNUMBER (LOC X)
        8 T NIL \NUMSTR \NUMSTR1)))
      (SYSTEMTYPEPUNT ' (\PRINDATUM X])

```

(PRINSTRING

```

[LAMBDA (X STREAM) ; (* bvm%: "11-May-86 15:08")
  (COND
    [*PRINT-ESCAPE* ; Print with double quotes and escaped as needed
     (LET ((ESC (fetch (READTABLEP ESCAPECHAR) of *READTABLE*)))
       [.SPACECHECK. STREAM (IPLUS 2 (\NSTRINGCHARS X)
         (for C instring X count (OR (EQ C (CHARCODE %"))

```

(EQ C ESC]

```

(\OUTCHAR STREAM (CHARCODE %"))
(for C instring X do (COND
  ((OR (EQ C (CHARCODE %"))
        (EQ C (CHARCODE LF))
        (EQ C ESC)) ; VM says only " is escaped no matter what stringdelim's are.
        (\OUTCHAR STREAM ESC)))
  (\OUTCHAR STREAM C))
(\OUTCHAR STREAM (CHARCODE %"))
(T (.SPACECHECK. STREAM (\NSTRINGCHARS X))
  (\SOUT X STREAM])

```

(\SOUT

[LAMBDA (X STREAM)

; Edited 14-Dec-88 22:17 by jds

;; Print the string X onto STREAM, which -must- be a stream.

```

(DECLARE (GLOBALVARS \DISPLAYSTREAMTYPES))
(DECLARE (SPECVARS *DRIBBLE-OUTPUT* \PRINTERMSA \TERM.OFD))
(COND
  [(FMEMB (ffetch (IMAGEOPS IMAGETYPE) of (fetch (STREAM IMAGEOPS) of STREAM))
    \DISPLAYSTREAMTYPES)
  (LET ((*DRIBBLE-OUTPUT* *DRIBBLE-OUTPUT*)
        (\PRINTERMSA \PRINTERMSA)
        (\TERM.OFD \TERM.OFD))
    (for I instring X do (\OUTCHAR STREAM I])
  ((for I instring X do (\OUTCHAR STREAM I])
)

```

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

```

(PUTPROPS .FILELINELENGTH. MACRO ((STRM)
  (LET ((L (fetch (STREAM LINELENGTH) of STRM)))
    (SELECTC L
      (0 \LINELENGTH) (* Some default)
      (MAX.SMALLP \NIL) (* Infinite)
      L)))
)

```

(DEFMACRO \PRINDATUM-LISTP ()

;; This is a hokey macro call to save the function call. Read it as though it were inline code in \prindatum

```

\LET (LABEL FIRSTTIME)
  (OR CPL (SETQ CPL 0))
  (if *PRINT-CIRCLE-HASHTABLE*
    then ;; *PRINT-CIRCLE-HASHTABLE* is only non-nil when *print-circle*.
      (CL:MULTIPLE-VALUE-SETQ (LABEL FIRSTTIME)
        (PRINT-CIRCLE-LOOKUP OBJECT)))
  [if LABEL
    then (\CKPOSSOUT STREAM LABEL)
    (CL:WHEN FIRSTTIME
      (\CKPOSBOUT STREAM (CHARCODE SPACE)))]
(COND
  ((AND LABEL (NOT FIRSTTIME)) ; Second reference --- just print label
  NIL)
  ((AND *PRINT-LEVEL* (ILEQ *PRINT-LEVEL* CPL))
  (\ELIDE.PRINT.ELEMENT STREAM))
  (T (PROG (CDRCNT)
    [COND
      (*PRINT-LENGTH* (SETQ CDRCNT (COND
        ((fetch (READTABLEP COMMONLISP) of *READTABLE*)
         0)
        (T ; Interlisp print depth is triangular, Common Lisp isn't
          [COND
            ((IGEQ CPL *PRINT-LENGTH*)
             ; We would just print '(-)' so it's nicer to print '&'
             (RETURN (\ELIDE.PRINT.ELEMENT STREAM)
              CPL]
             ; Recursive calls will be at 1 greater depth
            (add CPL 1)
            (\CKPOSBOUT STREAM (CHARCODE % ()))
            LP [COND
              ((AND CDRCNT (IGREATERP (add CDRCNT 1)
                *PRINT-LENGTH*)) ; have printed as many elements as allowed
              (\ELIDE.PRINT.TAIL STREAM T))
              (T (\PRINDATUM (CAR OBJECT)
                STREAM CPL)
              (COND
                ((LISTP (SETQ OBJECT (CDR OBJECT)))
                 (\CKPOSBOUT STREAM (CHARCODE SPACE))
                 (if (AND *PRINT-CIRCLE-HASHTABLE* (PRINT-CIRCLE-LABEL-P OBJECT))

```

```

      then ; Must print as a dotted tail
        (\CKPOSSOUT STREAM ". ")
        (\PRINDATUM OBJECT STREAM CPL)
      else (GO LP))) ; Dotted tail
(OBJECT (\CKPOSSOUT STREAM ". ")
         (\PRINDATUM OBJECT STREAM]
(\CKPOSBOUT STREAM (CHARCODE " "]))

```

:: FOLLOWING DEFINITIONS EXPORTED

```
(DECLARE%: EVAL@COMPILE
```

```
(PUTPROPS .SPACECHECK. MACRO ((STRM N)
                              (AND \THISFILELINELENGTH (IGREATERP (IPLUS N (fetch CHARPOSITION of STRM))
                              \THISFILELINELENGTH)
                              (FRESHLINE STRM))))

```

```
(PUTPROPS \CHECKRADIX MACRO [LAMBDA (R)
                             (COND
                              ((OR (NOT (SMALLP R))
                                   (ILESSP R 1)
                                   (IGREATERP R 36))
                               (\INVALID.RADIX R))
                              (T R])
                             )

```

:: END EXPORTED DEFINITIONS

```
(DEFINEQ
```

```
(\INVALID.RADIX
 [LAMBDA (N) ; (* bvm%: " 5-May-86 10:58")
  (ERROR "Bad value for *print-base*" N)]
)

```

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

```
(SPECVARS \THISFILELINELENGTH)
)

```

:: Internal printing

```
(DEFINEQ
```

```
(\MAPNAME
 [LAMBDA (FN X FLG RDTBL *PRINT-LEVEL* *PRINT-LENGTH*) ; Edited 23-Mar-87 11:01 by bvm:

```

::: Run thru the characters in the pname of X, calling FN on each character. For speed, FN is defined to be of the same form as an OUTCHARFN, viz.,
::: arglist = (stream char); stream in this case is a dummy

```

(LET [(*READTABLE* (if FLG
                      then (\GTREADTABLE RDTBL)
                      else (\DTEST *READTABLE* 'READTABLEP]
      (LET [(*PACKAGE* (if (AND FLG (fetch (READTABLEP USESILPACKAGE) of *READTABLE*))
                          then *INTERLISP-PACKAGE*
                          else *PACKAGE*))
            (*PRINT-ESCAPE* FLG)
            (*PRINT-BASE* (if (OR FLG PRXFLG)
                              then *PRINT-BASE*
                              else 10))
            (*PRINT-RADIX* (AND FLG *PRINT-RADIX*))])
      (\MAPNAME.INTERNAL FN X])

```

```
(\MAPNAME.INTERNAL
```

```

[LAMBDA (FN X) ; Edited 10-May-2021 15:46 by rmk:
  (WITH-RESOURCE (\MAPNAMESTREAM)

```

```

    (replace (STREAM OUTCHARFN) of \MAPNAMESTREAM with FN) ; Should never use the bout fn, but include it just in case
    (replace STRMBOUTFN of \MAPNAMESTREAM with FN) ; somebody thinks \OUTCHAR = \BOUT
    ; Stream has no linelength checks
    (LET (\THISFILELINELENGTH)
          (\DECLARE (SPECVARS \THISFILELINELENGTH))
          (\PRINDATUM X \MAPNAMESTREAM 0])

```

```
(\PNAMESTREAMP
```

```

[LAMBDA (STRM) ; (* bvm%: "24-Mar-86 17:37")

```

::: True if STRM is an internal-printing stream for pnames, i.e., one of the values of the \MAPNAMESTREAM resource

```

(AND (TYPENAMEP STRM 'STREAM)
      (EQ (fetch (STREAM DEVICE) of STRM)
           \PNAMEDEVICE])

```

```

)
(DECLARE%: DONTCOPY
(DECLARE%: EVAL@COMPILE
[PUTDEF '\MAPNAMESTREAM 'RESOURCES '(NEW (create STREAM DEVICE _ \PNAMEDEVICE ACCESSBITS _ OutputBits
LINELENGTH _ MAX.SMALLP]
)
(DECLARE%: EVAL@COMPILE
(PUTPROPS PNAMESTREAMAMP DMACRO ((STRM)
(EQ (fetch (STREAM DEVICE) of STRM)
\PNAMEDEVICE)))
)
)
(/SETTOPVAL '\MAPNAMESTREAM.GLOBALRESOURCE NIL)
(RPAQ? \PNAMEDEVICE (NCREATE 'FDEV (\GETDEVICEFROMHOSTNAME 'NULL T))
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \PNAMEDEVICE)
)

```

:: Obsolete

```

(DEFINEQ
(\MAPCHARS
[LAMBDA (\MAPCHARFN X FLG RDTBL) (* bvm%: "13-Mar-86 18:53")
(DECLARE (SPECVARS RDTBL))

```

::: Run thru the characters in the pname of X, calling \MAPCHARFN on each character.

```

(\MAPNAME [FUNCTION (LAMBDA (DUMMY CHAR)
(SPREADAPPLY* \MAPCHARFN CHAR]
X FLG RDTBL])
)
(DECLARE%: EVAL@COMPILE DOCOPY
(ADDTOVAR SYSSPECVARS *PRINT-BASE* *READ-BASE* *PRINT-RADIX* *PRINT-ESCAPE* *PRINT-CASE* *PRINT-GENSYM*
*PRINT-LEVEL* *PRINT-LENGTH* *PRINT-PRETTY* *PRINT-CIRCLE* *PRINT-ARRAY* *PACKAGE*
)
)

```

:: PRINTNUM and friends

```

(DEFINEQ
(PRINTNUM
[LAMBDA (FORMAT NUMBER FILE) (* DECLARATIONS%: (RECORD FIXFMT
(WIDTH RADIX PAD0 LEFTFLUSH))
(RECORD FLOATFMT (WIDTH DECPART EXPPART PAD0
SIGDIGITS)))
(* rmk%: "17-MAY-82 10:07")
(DECLARE (GLOBALVARS NILNUMPRINTFLG)
(GLOBALRESOURCE (\NUMSTR \NUMSTR1)
(PROG (STR WIDTH PAD TEMP RAD (FLOATFLAG (SELECTQ (CAR (LISTP FORMAT))
(FLOAT T)
(FIX NIL)
(LISPERROR "ILLEGAL ARG" FORMAT))))
(FMT (CDR FORMAT)))
(SETQ WIDTH (fetch WIDTH of FMT))
[SETQ STR (COND
((AND (NULL NUMBER)
NILNUMPRINTFLG))
(FLOATFLAG (\CONVERT.FLOATING.NUMBER (FLOAT NUMBER)
\NUMSTR \NUMSTR1 (\CHECKFLTFMT FORMAT)))
(T (\CONVERTNUMBER (OR (FIXP NUMBER)
(FIXR NUMBER))
(COND
((SETQ RAD (fetch RADIX of FMT))
(SETQ TEMP (IABS RAD))
(COND
((OR (IGREATERP 2 TEMP)
(IGREATERP TEMP 16))
(\ILLEGAL.ARG RAD)))
TEMP)
(T 10))
(OR (NULL RAD)

```

```

                                (IGREATERP RAD 0))
                                NIL \NUMSTR \NUMSTR1]
(SETQ PAD (COND
            (WIDTH (IDIFFERENCE WIDTH (NCHARS STR)))
            (T 0)))
[COND
  ((AND (IGREATERP PAD 0)
        (OR FLOATFLAG (NULL (fetch LEFTFLUSH of FMT))
            (COND
              ((COND
                (FLOATFLAG (fetch (FLOATFMT PAD0) of FMT))
                (T (fetch (FIXFMT PAD0) of FMT)))
              (FRPTQ PAD (PRIN1 "0" FILE)))
              (T (SPACES PAD FILE))
            (PRIN1 STR FILE)
            (COND
              ((AND (IGREATERP PAD 0)
                    (NOT FLOATFLAG)
                    (fetch LEFTFLUSH of FMT))
               (SPACES PAD FILE)))
            (RETURN NUMBER])

```

(FLTFMT

```

[LAMBDA (FORMAT)
  (PROG1 \FLOATFORMAT
    (AND FORMAT (\CHECKFLTFMT FORMAT)
      (SETQ \FLOATFORMAT FORMAT))))]
(* bvm%: "30-JAN-81 23:20")
; numeric arg, as on 10, not allowed

```

(\CHECKFLTFMT

```

[LAMBDA (FORMAT)
  (* bvm%: "29-JAN-81 15:41")

```

::: Generates error if FORMAT is not legal FLOAT format: (FLOAT WIDTH DECPART EXPPART PAD SIGDIGITS)

```

(COND
  ((OR (EQ FORMAT T)
        (AND (EQ (CAR FORMAT)
                  'FLOAT)
              (EVERY (CDR FORMAT)
                     (FUNCTION (LAMBDA (X)
                                (OR (NULL X)
                                    (FIXP X)
                                (FORMAT)
                                (T (LISPERROR "ILLEGAL ARG" FORMAT))

```

(PRINTNUM-TO-STRING

```

[LAMBDA (FORMAT NUMBER)
  (* DECLARATIONS%: (RECORD FIXFMT
                    (WIDTH RADIX PAD0 LEFTFLUSH))
    (RECORD FLOATFMT (WIDTH DECPART EXPPART PAD0
                      SIGDIGITS)))
  ; Edited 27-Nov-91 13:32 by jds

```

```

(DECLARE (GLOBALVARS NILNUMPRINTFLG))
(GLOBALRESOURCE (\NUMSTR \NUMSTR1)
  (PROG (STR WIDTH PAD TEMP RAD (FLOATFLAG (SELECTQ (CAR (LISTP FORMAT))
                                                    (FLOAT T)
                                                    (FIX NIL)
                                                    (LISPERROR "ILLEGAL ARG" FORMAT))))
        (FMT (CDR FORMAT)))
    (SETQ WIDTH (fetch WIDTH of FMT))
    [SETQ STR (COND
              ((AND (NULL NUMBER)
                    NILNUMPRINTFLG))
              (FLOATFLAG (\CONVERT.FLOATING.NUMBER (FLOAT NUMBER)
                                                    \NUMSTR \NUMSTR1 (\CHECKFLTFMT FORMAT)))
              (T (\CONVERTNUMBER (OR (FIXP NUMBER)
                                     (FIXR NUMBER))
                                (COND
                                  ((SETQ RAD (fetch RADIX of FMT))
                                   (SETQ TEMP (IABS RAD))
                                   (COND
                                    ((OR (IGREATERP 2 TEMP)
                                         (IGREATERP TEMP 16))
                                     (\ILLEGAL.ARG RAD)))
                                   TEMP)
                                  (T 10))
                                  (OR (NULL RAD)
                                      (IGREATERP RAD 0))
                                  NIL \NUMSTR \NUMSTR1]
              (SETQ PAD (COND
                        (WIDTH (IDIFFERENCE WIDTH (NCHARS STR)))
                        (T 0)))
              (RETURN (CONCAT (COND
                              [[AND (IGREATERP PAD 0)
                                (OR FLOATFLAG (NULL (fetch LEFTFLUSH of FMT]

```



```
(COND
  ((COND
    (FLOATFLAG (fetch (FLOATFMT PAD0) of FMT))
    (T (fetch (FIXFMT PAD0) of FMT)))
  (ALLOCSTRING PAD "0"))
  (T (ALLOCSTRING PAD " "]
    (T ""))
  STR])
)
(DECLARE%: EVAL@COMPILE
(PROGN (PUTPROPS NUMFORMATCODE BYTEMACRO (= . PROG1))
  (PUTPROPS NUMFORMATCODE DMACRO (= . PROG1)))
)
(RPAQ? NILNUMPRINTFLG )
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(LOCALVARS . T)
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS \LINELENGTH \FLOATFORMAT PRXFLG \DEFPRINTFNS)
)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS
(ADDTOVAR NLAMA )
(ADDTOVAR NLAML )
(ADDTOVAR LAMA )
)
(PUTPROPS APRINT COPYRIGHT ("Venue & Xerox Corporation" 1982 1983 1984 1985 1986 1987 1988 1990 1991 2020 2021)
)
```

FUNCTION INDEX

DEFPRINT	3	PRINT-CIRCLE-LOOKUP	4	\CONVERTNUMBER	7	\PRINDATUM	5
FLTFMT	16	PRINT-CIRCLE-SCAN	5	\ELIDE.ELEMENT.CHAR	6	\PRINSTACKP	12
FRESHLINE	3	PRINTCCODE	3	\ELIDE.PRINT.ELEMENT	6	\PRINSTRING	12
LINELENGTH	4	PRINTLEVEL	3	\ELIDE.PRINT.TAIL	6	\PRINT-USING-ADDRESS	6
PNAMESTREAMP	14	PRINTNUM	15	\ELIDE.TAIL.STRING	7	\PRINT-USING-DEFPRINT	6
PRIN1	1	PRINTNUM-TO-STRING	16	\INVALID.RADIX	14	\PRINTADDR	12
PRIN2	2	RADIX	3	\LITPRIN	8	\PRINTCCODE	3
PRIN3	2	SPACES	3	\LITPRIN.INTERNAL	8	\SOUT	13
PRIN4	2	TERPRI	3	\MAPCHARS	15	\SYMBOL.ESCAPE.COUNT	10
PRINT	2	\CHECKFLTFMT	16	\MAPNAME	14		
PRINT-CIRCLE-ENTER	5	\CKPOSBOUT	7	\MAPNAME.INTERNAL	14		
PRINT-CIRCLE-LABEL-P	4	\CKPOSSOUT	7	\NUMERIC.PNAMEP	11		

VARIABLE INDEX

INTERLISP-PRIN1-CASE	4	*PRINT-ESCAPE*	4	PLVLFLEFLG	4
KEYWORD-PACKAGE	4	*PRINT-GENSYM*	4	PRXFLG	4
PACKAGE	4	*PRINT-LENGTH*	4	SYSSPECVARS	15
PRINT-ARRAY	4	*PRINT-LEVEL*	4	\DEFPRINTFNS	4
PRINT-BASE	4	*PRINT-PRETTY*	4	\FLOATFORMAT	4
PRINT-CASE	4	*PRINT-RADIX*	4	\LINELENGTH	4
PRINT-CIRCLE	4	*READ-BASE*	4	\PNAMEDEVICE	15
PRINT-CIRCLE-HASHTABLE	4	NILNUMPRINTFLG	17		

MACRO INDEX

.FILELINELENGTH. .13	.SPACECHECK.14	PNAMESTREAMP	15	\CHECKRADIX	14	\PRINDATUM-LISTP .13
----------------------	---------------------	--------------------	----	-------------------	----	----------------------
