

File created: 11-Oct-93 16:04:23 {Pele:mv:envos}<LispCore>Sources>CLTL2>APRINT.;1

changes to: (VARS APRINTCOMS)
(FUNCTIONS \PRINDATUM-LISTP)

previous date: 4-Jan-92 15:40:50 {DSK}<mo>usr>users>sybalsky>cltl2>sources>APRINT.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1982, 1983, 1984, 1985, 1986, 1987, 1988, 1990, 1991, 1992, 1993 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **APRINTCOMS**

```
[ (COMS                                     ; User-level print functions
  (FNS PRIN1 PRIN2 PRIN3 PRIN4 PRINT PRINTCCODE PRINTLEVEL RADIX SPACES TERPRI FRESHLINE DEFPRINT
    LINELENGTH))
  (FNS ALPHANUMCHARARRAY)
  (INITVARS (PLVLFILFLG NIL)
    (\LINELENGTH 82)
    (\FLOATFORMAT T)
    (PRXFLG NIL)
    (*PRINT-BASE* 10)
    (*READ-BASE* 10)
    (*READ-EVAL* T)
    (*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)
    (*CLTL2-PEDANTIC* NIL)
    (\DEFPRINTFNS NIL)
    (ALPHANUMCHARARRAY (ALPHANUMCHARARRAY)))

  (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 \PNAME-CONTAINS-NO-EOL \SYMBOL.ESCAPE.COUNT \NUMERIC.PNAMEP
      \PRINSTACKP \PRINTADDR \PRINSTSTRING \SOUT \OUTCHAR)
    (FNS \FILEOUTCHARFN \JISFILEOUTCHARFN \SHIFTJISFILEOUTCHARFN \EUCFILEOUTCHARFN
      \THROUGHFILEOUTCHARFN)
    (DECLARE%: EVAL@COMPILE DONTCOPY (MACROS .FILELINELENGTH.)
      (FUNCTIONS \PRINDATUM-LISTP)
      (EXPORT (MACROS .SPACECHECK. \CHECKKRADIX)
        (MACROS \XCCSFILEOUTCHARFN)))
    (FNS \INVALID.RADIX)
    (SPECVARS \THISFILELINELENGTH))

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

  (COMS                                     ; Obsolete
    (FNS \MAPCHARS))
    (DECLARE%: EVAL@COMPILE DOCOPY
      (ADDVARS (SYSSPECVARS *PRINT-BASE* *READ-BASE* *READ-EVAL* *PRINT-RADIX* *PRINT-ESCAPE*
        *PRINT-CASE* *PRINT-GENSYM* *PRINT-LEVEL* *PRINT-LENGTH* *PRINT-PRETTY*
        *PRINT-CIRCLE* *PRINT-ARRAY* *PACKAGE* *CLTL2-PEDANTIC*)))

  (COMS                                     ; PRINTNUM and friends
    (FNS PRINTNUM FLTFMT \CHECKFLTFMT PRINTNUM-TO-STRING)
    (MACROS NUMFORMATCODE)
    (INITVARS (NILNUMPRINTFLG)))
    (LOCALVARS . T)
    (GLOBALVARS \LINELENGTH \FLOATFORMAT PRXFLG \DEFPRINTFNS ALPHANUMCHARARRAY)
    (PROP (FILETYPE APRINT))
    (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVARS (ADDVARS (NLAMA)
      (NLAML)
      (LAMA]))

  (COMS                                     ; 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* 'READTABLEP))
                           (OR (\OUTERMP STRM)
                               PLVLFILEFLG])
      (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 prin1
              ; 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)
                               PLVLFILEFLG])
      (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)
                               PLVLFILEFLG])
      (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)
                               PLVLFILEFLG])
      (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)
  (LET [(STRM (\GETSTREAM FILE 'OUTPUT)
    (PRIN2 X STRM RDTBL)
    (\OUTCHAR STRM (CHARCODE EOL))
    X])]
```

(* bvm%: " 9-May-86 23:08")

(PRINTCCODE

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

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

(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)
```

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

;; 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])]
```

(DEFPRINT

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

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

; The FIXP case should never occur

(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)))))]])
)
```

(DEFINEQ

(ALPHANUMCHARARRAY

[LAMBDA NIL

; Edited 19-Sep-91 19:46 by jrb:

```

(SETQ ALPHANUMCHARARRAY (ARRAY 256 'BYTE 0 0))
(for I from (CHARCODE A) to (CHARCODE Z) do (SETA ALPHANUMCHARARRAY I 1))
(for I from (CHARCODE a) to (CHARCODE z) do (SETA ALPHANUMCHARARRAY I 2))
(for I from (CHARCODE 0) to (CHARCODE 9) do (SETA ALPHANUMCHARARRAY I 3))
ALPHANUMCHARARRAY])
)
```

(RPAQ? **PLVLFILEFLG** NIL)(RPAQ? **\LINELENGTH** 82)(RPAQ? **\FLOATFORMAT** T)(RPAQ? **PRXFLG** NIL)(RPAQ? ***PRINT-BASE*** 10)(RPAQ? ***READ-BASE*** 10)(RPAQ? ***READ-EVAL*** T)(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? ***CLTL2-PEDANTIC*** NIL)(RPAQ? **\DEFPRINTFNS** NIL)(RPAQ? **ALPHANUMCHARARRAY** (ALPHANUMCHARARRAY))

;; 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

```

[CL:LAMBDA (OBJECT)                                     ; Edited 3-Apr-91 10:33 by jrb:
  (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:SYMBOL
            (CL:UNLESS (ffetch (CL:SYMBOL PACKAGE) of OBJECT) ; Uninterned symbols must be entered
                        (PRINT-CIRCLE-ENTER 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

```

[CL:LAMBDA (OBJECT)
  (DECLARE (CL:SPECIAL *PRINT-CIRCLE-HASHTABLE* THERE-ARE-CIRCLES))
  ; Edited 31-Mar-87 19:16 by jrb:
  (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!")))))

```

)