

File created: 23-Sep-2024 11:55:33 {DSK}<home>matt>Interlisp>medley>sources>CMLREAD.;4

edit by: mth

changes to: (FNS CL:READ-FROM-STRING)

previous date: 16-Sep-2024 12:26:09 {DSK}<home>matt>Interlisp>medley>sources>CMLREAD.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(RPAQQ CMLREADCOMS

```
[ (COMS
  ;; Misc Common Lisp reader functions
  (FNS CL:COPY-READTABLE)
  (FNS CL:READ-LINE CL:READ-CHAR CL:UNREAD-CHAR CL:PEEK-CHAR CL:LISTEN CL:READ-CHAR-NO-HANG
    CL:CLEAR-INPUT CL:READ-FROM-STRING CL:READ-BYTE CL:WRITE-BYTE)
    ; must turn off packed version of CLISP infix
  (VARS [CLISPCHARS (LDIFFERENCE CLISPCHARS '(- *)
    (CLISPCHARRAY (MAKEBITTABLE CLISPCHARS))
    (DWIMINMACROSFLG))
  (VARIABLES *READ-DEFAULT-FLOAT-FORMAT*)
  (GLOBALVARS CMLRDTBL READ-LINE-RDTBL))
[COMS
  ;; Crude means to aid reading and printing things in same reader environment. There are some fns and an INITRECORDS for this on
  ;; ATBL to get it early in the loadup
  (RECORDS READER-ENVIRONMENT)
  (FUNCTIONS WITH-READER-ENVIRONMENT)
  (ADDVARS (SYSSPECVARS *PACKAGE* *READTABLE* *READ-BASE* *PRINT-BASE*))
  (PROP INFO WITH-READER-ENVIRONMENT)
  (GLOBALVARS *COMMON-LISP-READ-ENVIRONMENT*)
  (INITVARS (*COMMON-LISP-READ-ENVIRONMENT* (create READER-ENVIRONMENT REPACKAGE _ (CL:FIND-PACKAGE
    "USER")
    REREADTABLE _ CMLRDTBL REBASE _ 10 REFORMAT _
    :XCCS]
  (PROP FILETYPE CMLREAD)
  (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
    (ADDVARS (NLAMA)
      (NLAML)
      (LAMA CL:READ-BYTE CL:READ-FROM-STRING CL:CLEAR-INPUT CL:READ-CHAR-NO-HANG CL:PEEK-CHAR
        CL:UNREAD-CHAR CL:READ-CHAR CL:READ-LINE CL:COPY-READTABLE]))

```

;; Misc Common Lisp reader functions

(DEFINEQ

(CL:COPY-READTABLE

```
[CL:LAMBDA (&OPTIONAL (FROM-READTABLE *READTABLE*)
  TO-READTABLE)
  (* bvm%: "13-Oct-86 15:21")
  ; If FROM-READTABLE is NIL, then a copy of a standard
  ; Common Lisp readtable is made.
  (if (AND (NULL FROM-READTABLE)
    (NULL TO-READTABLE))
    then
      ; just make a brand new one
      (CMLRDTBL)
    else (SETQ FROM-READTABLE (\DTEST (OR FROM-READTABLE (CMLRDTBL))
      'READTABLEP))
      (if TO-READTABLE
        then (RESETREADTABLE (\DTEST TO-READTABLE 'READTABLEP)
          FROM-READTABLE)
        TO-READTABLE
      else (COPYREADTABLE FROM-READTABLE))
)
```

(DEFINEQ

(CL:READ-LINE

```
[CL:LAMBDA (&OPTIONAL STREAM (EOF-ERRORP T)
  EOF-VALUE RECURSIVE-P)
  ; Edited 31-Mar-87 18:36 by bvm:
  ;; Returns a line of text read from the STREAM as a string, discarding the newline character.
  (CL:SETQ STREAM (\GETSTREAM STREAM 'INPUT))
  (if (AND (NULL EOF-ERRORP)
    (NULL RECURSIVE-P)
    (\EOFP STREAM))
    then EOF-VALUE
    else (LET ((RESULT (RSTRING STREAM READ-LINE-RDTBL)))
      (if (\EOFP STREAM)
        then (CL:VALUES RESULT T)
        else
          ; consume the eol
          (READCCODE STREAM)
          (CL:VALUES RESULT NIL]))
)
```

(CL:READ-CHAR

```

[CL:LAMBDA (&OPTIONAL (STREAM *STANDARD-INPUT*)
            (EOF-ERRORP T)
            EOF-VALUE RECURSIVE-P)
  ;; Inputs a character from STREAM and returns it.
  (LET [(STREAM (\GETSTREAM STREAM 'INPUT))
        (COND
         ((AND (NOT EOF-ERRORP)
               (NOT RECURSIVE-P)
               (\EOFP STREAM))
          EOF-VALUE)
         (T (CL:CODE-CHAR (READCCODE STREAM)))]

```

; Edited 14-Dec-86 20:41 by bvm:

(CL:UNREAD-CHAR

```

[CL:LAMBDA (CHARACTER &OPTIONAL (INPUT-STREAM *STANDARD-INPUT*))

```

; Edited 23-Jun-2021 13:05 by rmk:

;; Puts the CHARACTER back on the front of the input STREAM. According to the manual, 'One may apply UNREAD-CHAR only to the character most recently read from INPUT-STREAM.'

```

(\BACKCCODE (\GETSTREAM INPUT-STREAM 'INPUT))
NIL))

```

(CL:PEEK-CHAR

```

[CL:LAMBDA (&OPTIONAL (PEEK-TYPE NIL)
            (STREAM *STANDARD-INPUT*)
            (EOF-ERRORP T)
            EOF-VALUE RECURSIVE-P)

```

; Edited 19-Jul-2022 23:29 by rmk;
; Edited 14-Apr-87 14:39 by bvm:

;; Peeks at the next character in the input Stream. See manual for details.

```

(DECLARE (IGNORE RECURSIVE-P))
(LET ((STREAM (\GETSTREAM STREAM 'INPUT))
      (\RefillBufferFn '\PEEKREFILL)
      CL:CHAR)
  (DECLARE (CL:SPECIAL \RefillBufferFn))
  (SELECTQ PEEK-TYPE
    (NIL

```

; standard case--return next char. \peekccode to terminal
; requires the binding of \RefillBufferFn above

```

      (if (SETQ CL:CHAR (\PEEKCCODE.EOLC STREAM (NULL EOF-ERRORP)))
          then (CL:CODE-CHAR CL:CHAR)
          else EOF-VALUE))

```

; skip whitespace before peeking

```

      (T
       (if (SETQ CL:CHAR (SKIPSEPRCODES STREAM))
           then (CL:CODE-CHAR CL:CHAR)
           elseif EOF-ERRORP
              then (\EOF.ACTION STREAM)
           else EOF-VALUE))

```

```

      (if (CL:CHARACTERP PEEK-TYPE)
          then (LET ((DESIREDCHAR (CL:CHAR-CODE PEEK-TYPE))
                    (NOERROR (NULL EOF-ERRORP)))
                (until (EQ (SETQ CL:CHAR (\PEEKCCODE.EOLC STREAM NOERROR))
                          DESIREDCHAR)
                      do (if (NULL CL:CHAR)
                            then (RETURN EOF-VALUE))))

```

;; READCCODE sets STREAM's LASTCCODE, \INCCODE.EOLC doesn't

```

          (READCCODE STREAM)
          finally (RETURN PEEK-TYPE)))
      else (\ILLEGAL.ARG PEEK-TYPE)]

```

(CL:LISTEN

```

[CL:LAMBDA (&OPTIONAL STREAM)

```

; Edited 14-Apr-87 16:49 by bvm:

;; Returns T if a character is available on the given STREAM

```

(READP (\GETSTREAM STREAM 'INPUT)
T)))

```

(CL:READ-CHAR-NO-HANG

```

[CL:LAMBDA (&OPTIONAL STREAM (EOF-ERRORP T)
            EOF-VALUE RECURSIVE-P)

```

; Edited 14-Apr-87 16:40 by bvm:

;; Returns the next character from the STREAM if one is available, or NIL. However, if STREAM is at eof, do eof handling.

```

(COND
  ((READP STREAM T)
   (CL:READ-CHAR STREAM EOF-ERRORP EOF-VALUE RECURSIVE-P))
  ((NOT (EOFP STREAM))
   NIL)
  (EOF-ERRORP (\EOF.ACTION STREAM))
  (T EOF-VALUE)))

```

; there is input, get it
; there could be more input, so don't wait, return NIL

(CL: CLEAR-INPUT

```
[CL:LAMBDA (&OPTIONAL (STREAM *STANDARD-INPUT*))
  ;; Clears any buffered input associated with the Stream.
  (CLEARBUF (\GETSTREAM STREAM 'INPUT])
  (* bvm%: "13-Oct-86 15:46")
```

(CL: READ-FROM-STRING

```
[CL:LAMBDA (STRING &OPTIONAL EOF-ERROR-P EOF-VALUE &KEY START END PRESERVE-WHITESPACE)
  ; Edited 23-Sep-2024 11:47 by mth
  ; Edited 16-Sep-2024 12:22 by mth
  ; Edited 8-Jun-90 14:15 by ymasuda
  (LET [(STREAM (OPENSTRINGSTREAM (COND
    [END (SUBSTRING STRING 1 (IMIN END (NCHARS STRING)
      (T (MKSTRING STRING)
    [COND
      (START (SETFILEPTR STREAM (UNFOLD START 2)
      (CL:VALUES (CL:IF PRESERVE-WHITESPACE
        (CL:READ-PRESERVING-WHITESPACE STREAM EOF-ERROR-P EOF-VALUE)
        (CL:READ STREAM EOF-ERROR-P EOF-VALUE))
      (FOLDLO (\GETFILEPTR STREAM
        2])
```

(CL: READ-BYTE

```
[CL:LAMBDA (BINARY-INPUT-STREAM &OPTIONAL (EOF-ERRORP T)
  EOF-VALUE)
  (* bvm%: "13-Oct-86 15:49")
  ;; Returns the next byte of the BINARY-INPUT-STREAM
  (LET [(STREAM (\GETSTREAM BINARY-INPUT-STREAM 'INPUT)
    (CL:IF (AND (NOT EOF-ERRORP)
      (\EOFP STREAM))
      EOF-VALUE
      (\BIN STREAM))])
```

(CL: WRITE-BYTE

```
(CL:LAMBDA (INTEGER BINARY-OUTPUT-STREAM)
  (* bvm%: "13-Oct-86 15:49")
  ;; Outputs the INTEGER to the binary BINARY-OUTPUT-STREAM
  (BOUT BINARY-OUTPUT-STREAM INTEGER)
  INTEGER))
```

;; must turn off packed version of CLISP infix

```
(RPAQ CLISPCHARS (LDIFFERENCE CLISPCHARS '(- *)))
(RPAQ CLISPCHARRAY (MAKEBITTABLE CLISPCHARS))
(RPAQQ DWIMINMACROSFLG NIL)
```

```
(CL:DEFVAR *READ-DEFAULT-FLOAT-FORMAT* 'CL:SINGLE-FLOAT)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS CMLRDTBL READ-LINE-RDTBL)
)
```

;; Crude means to aid reading and printing things in same reader environment. There are some fns and an INITRECORDS for this on ATBL to get it
;; early in the loadup

```
(DECLARE%: EVAL@COMPILE
(DATATYPE READER-ENVIRONMENT (REPACKAGE REREADTABLE REBASE REPACKAGEFORM REFORMAT REREADTABLEFORM))
)
(/DECLAREDATATYPE 'READER-ENVIRONMENT ' (POINTER POINTER POINTER POINTER POINTER POINTER)
  ;; ---field descriptor list elided by lister---
  ' 12)
```

(DEFMACRO WITH-READER-ENVIRONMENT (ENV . BODY)

```
`((CL:LAMBDA (E)
  (LET ((*PACKAGE* (ffetch (READER-ENVIRONMENT REPACKAGE) of E))
    (*READTABLE* (ffetch (READER-ENVIRONMENT REREADTABLE) of E))
    (*READ-BASE* (ffetch (READER-ENVIRONMENT REBASE) of E))
    (*PRINT-BASE* (ffetch (READER-ENVIRONMENT REBASE) of E)))
    ,@BODY))
  (\DTEST ,ENV 'READER-ENVIRONMENT)))
```

```
(ADDOVAR SYSSPECVARS *PACKAGE* *READTABLE* *READ-BASE* *PRINT-BASE*)
```

```
{MEDLEY}<sources>CMLREAD.;1

(PUTPROPS WITH-READER-ENVIRONMENT INFO EVAL)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS *COMMON-LISP-READ-ENVIRONMENT*)
)

(RPAQ? *COMMON-LISP-READ-ENVIRONMENT* (create READER-ENVIRONMENT REPACKAGE _ (CL:FIND-PACKAGE "USER")
                                             REREADTABLE _ CMLRDTBL REBASE _ 10 REFORMAT _ :XCCS))

(PUTPROPS CMLREAD FILETYPE CL:COMPILE-FILE)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA CL:READ-BYTE CL:READ-FROM-STRING CL:CLEAR-INPUT CL:READ-CHAR-NO-HANG CL:PEEK-CHAR CL:UNREAD-CHAR
           CL:READ-CHAR CL:READ-LINE CL:COPY-READTABLE)
)
```

FUNCTION INDEX

CL: CLEAR-INPUT3 CL: PEEK-CHAR2 CL: READ-CHAR-NO-HANG2 CL: UNREAD-CHAR2
CL: COPY-READTABLE1 CL: READ-BYTE3 CL: READ-FROM-STRING3 CL: WRITE-BYTE3
CL: LISTEN2 CL: READ-CHAR2 CL: READ-LINE1

VARIABLE INDEX

COMMON-LISP-READ-ENVIRONMENT ...4 CLISPCHARRAY3 DWIMINMACROFLG3
READ-DEFAULT-FLOAT-FORMAT3 CLISPCHARS3 SYSSPECVARS3

PROPERTY INDEX

CMLREAD4 WITH-READER-ENVIRONMENT4

MACRO INDEX

WITH-READER-ENVIRONMENT3

RECORD INDEX

READER-ENVIRONMENT3
