

File created: 23-Feb-88 11:13:23 {ERINYES}<LISPUSERS>LYRIC>PP-CODE-FILE.;1

changes to: (IL:FUNCTIONS PP-CODE-FILE-INTERNAL)

previous date: 3-Nov-87 12:26:37 {IE:PARC:XEROX}<LISP>LYRIC>LISPUSERS>PP-CODE-FILE.;2

Read Table: XCL

Package: XEROX-COMMON-LISP

Format: XCCS

; Copyright (c) 1987, 1988 by Xerox Corporation. All rights reserved.

```
(IL:RPAQQ IL:PP-CODE-FILECOMS
  ((IL:PROP (IL:MAKEFILE-ENVIRONMENT IL:FILETYPE)
    IL:PP-CODE-FILE)
    (IL:FUNCTIONS PP-CODE-FILE PP-CODE-FILE-INTERNAL FILE-MANAGER-FILE-P MAYBE-PP-CODE-FILE
      PRETTY-LISTFILES1)
    (IL:COMMANDS "see")
    (IL:P (IL:MOVD? 'IL:LISTFILES1 'IL:LISTFILES1-ORIGINAL)
      (IL:/MOVD 'PRETTY-LISTFILES1 'IL:LISTFILES1)
      (IL:CHANGENAME 'IL:FB.FASTSEE.ONEFILE 'IL:PFCOPYBYTES 'MAYBE-PP-CODE-FILE))
    (IL:COMS (IL:FNS IL:SUPERPRINT/COMMENT)
      (IL:DECLARE\ : IL:EVAL@COMPILE IL:DONTCOPY (IL:FILES (IL:LOADCOMP)
        IL:NEWPRINTDEF))))))

(IL:PUTPROPS IL:PP-CODE-FILE IL:MAKEFILE-ENVIRONMENT (:READTABLE "XCL" :PACKAGE "XCL"))

(IL:PUTPROPS IL:PP-CODE-FILE IL:FILETYPE :COMPILE-FILE)

(DEFUN PP-CODE-FILE (CODE-FILE &OPTIONAL (OUTPUT *STANDARD-OUTPUT*)
  (READER-ENV (FILE-MANAGER-FILE-P CODE-FILE)))
  "Pretty print contents of file manager file"
  (DECLARE (SPECIAL IL:*OLD-INTERLISP-READ-ENVIRONMENT*))
  (IF READER-ENV
    (LET ((IN-STREAM (IF (STREAMP CODE-FILE)
      CODE-FILE
      (OPEN CODE-FILE :DIRECTION :INPUT))))
      (UNWIND-PROTECT
        (LET ((OUT-STREAM (COND
          ((STREAMP OUTPUT)
            OUTPUT)
          ((IL:WINDOWP OUTPUT)
            (IL:GETSTREAM OUTPUT))
          (T (OPEN OUTPUT :DIRECTION :OUTPUT :IF-EXISTS :NEW-VERSION))))
          (ABORT T))
          (UNWIND-PROTECT
            (IL:WITH-READER-ENVIRONMENT READER-ENV
              (UNLESS (EQ READER-ENV IL:*OLD-INTERLISP-READ-ENVIRONMENT*)
                ;; if FILE-MANAGER-FILE-P read a IL:DEFINE-FILE-INFO expression to get the reader environment then
                ;; we have to both print one to the output as well as read this one again.
                (IL:PRINT-READER-ENVIRONMENT READER-ENV OUT-STREAM)
                (TERPRI OUT-STREAM)
                (IL:WITH-READER-ENVIRONMENT IL:*OLD-INTERLISP-READ-ENVIRONMENT* (READ IN-STREAM
                  )))
                (PP-CODE-FILE-INTERNAL IN-STREAM OUT-STREAM)
                (SETQ ABORT NIL)
                (PATHNAME OUT-STREAM))
              (UNLESS (OR (STREAMP OUTPUT)
                (IL:WINDOWP OUTPUT))
                (CLOSE OUT-STREAM :ABORT ABORT))))
            (UNLESS (STREAMP CODE-FILE)
              (CLOSE IN-STREAM))))
          (ERROR "~S not a File Manager file" CODE-FILE)))

(DEFUN PP-CODE-FILE-INTERNAL (IL:IN-STREAM IL:OUT-STREAM)
  ;; presume read environment has been set up for us
  ;; we just need to pretty print from IN-STREAM to OUT-STREAM
  ;; i can write this much easier in interlisp...
  (IL:BIND (IL:**COMMENT**FLG IL:_ NIL)
    (IL:*PRINT-SEMICOLON-COMMENTS* IL:_ T)
    (IL:*DIVIDE-LONG-STRINGS* IL:_ T)
    (IL:PRETTYFLG IL:_ T)
    IL:NAMES IL:SEXP DECLARE (IL:SPECVARS IL:**COMMENT**FLG IL:*PRINT-SEMICOLON-COMMENTS*
      IL:*DIVIDE-LONG-STRINGS* IL:PRETTYFLG)
    IL:EACHTIME (IL:SKIPSEPRS IL:IN-STREAM *READTABLE*) IL:UNTIL (IL:EOFP IL:IN-STREAM)
    IL:DO ; read an expression
    (IL:SETQ IL:SEXP (IL:READ IL:IN-STREAM *READTABLE*))
    (COND
      ((AND (NULL IL:NAMES)
        (IL:LISTP IL:SEXP)
        (EQ (CAR IL:SEXP)
```

```

      'IL:RPAQQ)
      (IL:STREQUAL (IL:SUBSTRING (CADR IL:SEXP)
                                -4)
        "COMS"))
;; found the COMS
(LET ((IL:COMS (CADDR IL:SEXP)))
  ;; pull out the function names
  (IL:SETQ IL:NAMES (IL:APPEND (IL:INFILECOMS? NIL 'IL:FNS IL:COMS)
                              (IL:INFILECOMS? NIL 'IL:FUNCTIONS IL:COMS)))))

;; pretty print the expression
(IF (EQ (CAR IL:SEXP)
      'IL:DEFINEQ)
    (PROGN ;; print blank lines between DEFINEQ defs
      (FORMAT IL:OUT-STREAM "~S~%" (CAR IL:SEXP))
      (DOLIST (IL:DEF (CDR IL:SEXP))
        (IL:TERPRI IL:OUT-STREAM)
        (IL:PRINTDEF IL:DEF NIL (AND (IL:LISTP IL:SEXP)
                                     (EQ (CAR IL:SEXP)
                                         'IL:DEFINEQ))
          NIL IL:NAMES IL:OUT-STREAM)
        (IL:TERPRI IL:OUT-STREAM))
      (FORMAT IL:OUT-STREAM "~%" (CAR IL:SEXP)))
    (IL:PRINTDEF IL:SEXP NIL NIL NIL IL:NAMES IL:OUT-STREAM))
;; leave a blank line between each
(IL:TERPRI IL:OUT-STREAM))

(DEFUN FILE-MANAGER-FILE-P (FILE)
  ;; Returns NIL or a reader environment.
  (DECLARE (SPECIAL IL:*OLD-INTERLISP-READ-ENVIRONMENT*))
  (WITH-OPEN-STREAM (STREAM (OPEN FILE :DIRECTION :INPUT))
    (IL:WITH-READER-ENVIRONMENT IL:*OLD-INTERLISP-READ-ENVIRONMENT*
      (AND (EQL #\ (PEEK-CHAR T STREAM NIL NIL))
        (LET ((DEFINE-FILE-INFO (CAR (IL:NLSAQ (READ STREAM)))))
          (IF (CONSP DEFINE-FILE-INFO)
              (CASE (CAR DEFINE-FILE-INFO)
                ((IL:DEFINE-FILE-INFO (IL:\\DO-DEFINE-FILE-INFO NIL (CDR DEFINE-FILE-INFO)))
                 (IL:FILECREATED IL:*OLD-INTERLISP-READ-ENVIRONMENT*)))))
        ))))

(DEFUN MAYBE-PP-CODE-FILE (INPUT &OPTIONAL (OUTPUT *STANDARD-OUTPUT*))
  (LET ((READER-ENV (FILE-MANAGER-FILE-P INPUT))
        (IF READER-ENV
            (PP-CODE-FILE INPUT OUTPUT READER-ENV)
            (LET ((IN-STREAM (IF (STREAMP INPUT)
                                INPUT
                                (OPEN INPUT :DIRECTION :INPUT))))
              (UNWIND-PROTECT
                (IL:COPYCHARS IN-STREAM (IL:GETSTREAM OUTPUT 'IL:OUTPUT))
                (UNLESS (STREAMP INPUT)
                  (CLOSE IN-STREAM)))))))

(DEFUN PRETTY-LISTFILES1 (FILE OPTIONS)
  ;; MOVD'd onto IL:LISTFILES1.
  (LET ((READER-ENV (FILE-MANAGER-FILE-P FILE))
        (IF READER-ENV
            (LET* ((PATHNAME (PROBE-FILE FILE))
                  (NAMESTRING (NAMESTRING PATHNAME))
                  (TEMP-FILE 'NIL))
              (DECLARE (GLOBAL IL:DEFAULTPRINTERTYPE))
              (WITH-OPEN-STREAM (PRINT-STREAM (IL:OPENIMAGESTREAM "{LPT}" IL:DEFAULTPRINTERTYPE))
                (PP-CODE-FILE PATHNAME PRINT-STREAM)
                (IL:STREAMPROP PRINT-STREAM 'IL:PRINTOPTIONS (LIST* 'IL:DOCUMENT.NAME
                                                                      (OR (IL:LISTGET OPTIONS
                                                                      'IL:DOCUMENT.NAME)
                                                                      NAMESTRING)
                                                                      'IL:DOCUMENT.CREATION.DATE
                                                                      (IL:GETFILEINFO PATHNAME
                                                                      'IL:ICREATIONDATE)
                                                                      'IL:HEADING
                                                                      (OR (IL:LISTGET OPTIONS
                                                                      'IL:HEADING)
                                                                      (IL:CONCAT NAMESTRING " "
                                                                      (IL:GETFILEINFO PATHNAME
                                                                      'IL:CREATIONDATE))))
                                                                      OPTIONS))))
            (IF (IL:LISTGET OPTIONS 'IL:DELETE)
                (DELETE-FILE PATHNAME)))

```

;; not a code file -- punt

(IL:LISTFILES1-ORIGINAL FILE OPTIONS)))

```
(DEFCOMMAND "see" (IL:FILE) "print the contents of FILE on the screen"
  (MAYBE-PP-CODE-FILE IL:FILE))
```

(IL:MOVD? 'IL:LISTFILES1 'IL:LISTFILES1-ORIGINAL)

(IL:/MOVD 'PRETTY-LISTFILES1 'IL:LISTFILES1)

(IL:CHANGENAME 'IL:FB.FASTSEE.ONEFILE 'IL:PFCOPYBYTES 'MAYBE-PP-CODE-FILE)

(IL:DEFINEQ

(IL:SUPERPRINT/COMMENT

(IL:LAMBDA (IL:L IL:FILE)

; Edited 2-Nov-87 14:13 by drc:

(COND

```
((AND IL:**COMMENT**FLG (NOT IL:FILEFLG)
  (NOT IL:MAKEMAP))
```

```
; If:
; There's a shorthand for comments, and
; We're not printing to a file, and
; Ww're not making the file map, then
```

;; Print out the shorthand version of the comment, watching out for overflowing the current line.

(COND

```
((> (+ (IL:DSPXPOSITION NIL IL:FILE)
  (IL:STRINGWIDTH IL:**COMMENT**FLG IL:FILE))
  (IL:DSPRIGHTMARGIN NIL IL:FILE))
  (IL:PRINENDLINE (IL:DSPLEFTMARGIN NIL IL:FILE)
    IL:FILE)))
```

(IL:PRIN1S IL:**COMMENT**FLG NIL IL:FILE))

(T (PROG (IL:COMMENT-LMARGIN IL:COMMENT-RMARGIN IL:RIGHTFLG IL:FLUSH-LEFTP IL:SEMIP IL:BODY)

(COND

```
((IL:SETQ IL:RIGHTFLG (NOT (OR (IL:SUPERPRINTEQ (CADR IL:L)
  IL:COMMENTFLG)
```

(COND

```
((IL:SETQ IL:SEMIP (IL:SEMI-COLON-COMMENT-P IL:L))
  ; Only 1-semi comments go in right margin
```

(IL:NEQ IL:SEMIP 1))

(T ; use size heuristic

```
(> (IL:LENGTH IL:L)
  10))))))
```

; Print comment in the righthand margin

```
(IL:SETQ IL:COMMENT-LMARGIN (OR IL:COMMENTCOL (IL:SUPERPRINT/COMMENT1 IL:L IL:RMARGIN
  IL:FILE)))
```

(IL:SETQ IL:COMMENT-RMARGIN IL:RMARGIN))

```
(AND (EQ IL:SEMIP 3)
  (NOT IL:MAKEMAP))
```

```
; Comment should be printed flush left. Don't do this with DEdit
; lest we confuse it
```

(IL:SETQ IL:COMMENT-LMARGIN (IL:DSPLEFTMARGIN NIL IL:FILE))

(IL:SETQ IL:COMMENT-RMARGIN IL:RMARGIN))

```
(AND (EQ IL:SEMIP 2)
  (NOT IL:MAKEMAP))
```

; indent like code

```
(IL:SETQ IL:COMMENT-LMARGIN (MIN IL:LEFT (+ (IL:DSPLEFTMARGIN NIL IL:FILE)
  (IL:IQUOTIENT (- IL:RMARGIN (IL:DSPLEFTMARGIN
    NIL IL:FILE))
    3))))
```

(IL:SETQ IL:COMMENT-RMARGIN IL:RMARGIN))

(T ; Print comment centered and wide

(IL:SETQ IL:COMMENT-LMARGIN (IL:FIXR (IL:TIMES 0.1 IL:RMARGIN)))

(IL:SETQ IL:COMMENT-RMARGIN (- IL:RMARGIN IL:COMMENT-LMARGIN))

(COND

```
((EQ IL:COMMENT-LMARGIN (IL:DSPXPOSITION NIL IL:FILE))
```

```
;; HACK: Almost certainly called from REPP, so we must suppress the normal leading and trailing blank lines as they
;; have already been done
```

(IL:SETQ IL:RIGHTFLG T))))

(COND

(NULL IL:RIGHTFLG)

(IL:PRINENDLINE IL:COMMENT-LMARGIN IL:FILE))

(< IL:COMMENT-LMARGIN (IL:DSPXPOSITION NIL IL:FILE))

(IL:PRINENDLINE IL:COMMENT-LMARGIN IL:FILE))

(T (IL:DSPXPOSITION IL:COMMENT-LMARGIN IL:FILE)))

(IL:SETFONT (PROG1 (IL:SETFONT IL:COMMENTFONT IL:FILE)

(COND

```
((AND IL:SEMIP (NOT IL:MAKEMAP)
  (IL:STRINGP (IL:SETQ IL:BODY (CAR (IL:LISTP (CDR (IL:LISTP (CDR IL:L))))
```

```
))))
```

(NULL (CDDDR IL:L))

(OR (IL:IMAGESTREAMP IL:FILE)

IL:*PRINT-SEMICOLON-COMMENTS*))

; do nice semi-colon stuff

```
(IL:PRIN2-LONG-STRING IL:BODY IL:FILE NIL NIL IL:COMMENT-LMARGIN
  IL:COMMENT-RMARGIN T IL:SEMIP))
```

(T (IL:SUPERPRINT/COMMENT2 IL:L IL:COMMENT-LMARGIN (IL:IQUOTIENT (+

IL:COMMENT-LMARGIN

IL:COMMENT-RMARGIN
)

2)

IL:COMMENT-RMARGIN IL:FILE))))

IL:FILE)

(COND

((AND (OR (AND IL:SEMIP (NOT IL:MAKEMAP))

(NOT IL:RIGHTFLG))

(NOT (= (IL:DSPXPOSITION NIL IL:FILE)

(IL:DSPLEFTMARGIN NIL IL:FILE))))

;; AR 8475 JDS 4/16/87: If there's a semi-colon comment on this line, and we're not making the file map (??), and
;; RIGHTFLG is NIL (whatever that means) then force a new line.

(IL:PRINENDLINE (IL:DSPLEFTMARGIN NIL IL:FILE)

IL:FILE)))

; (OR RIGHTFLG (PRINENDLINE 0 FILE))

(RETURN IL:L))))))

)

(IL:DECLARE\ : IL:EVAL@COMPILE IL:DONTCOPY

(IL:FILESLOAD (IL:LOADCOMP)

IL:NEWPRINTDEF)

)

(IL:PUTPROPS IL:PP-CODE-FILE IL:COPYRIGHT ("Xerox Corporation" 1987 1988))

FUNCTION INDEX

FILE-MANAGER-FILE-P	2	PP-CODE-FILE	1	PRETTY-LISTFILES1	2
MAYBE-PP-CODE-FILE	2	PP-CODE-FILE-INTERNAL	1	IL: SUPERPRINT/COMMENT	3

COMMAND INDEX

"see"	3
-------------	---

PROPERTY INDEX

IL:PP-CODE-FILE	1
-----------------------	---
