

File created: 8-Feb-2023 16:21:26 {DSK}<home>larry>il>medley>sources>PRETTY.;3

edit by: lmm

changes to: (FNS PRINTDATE1)

previous date: 19-Jan-2022 20:35:18 {DSK}<home>larry>il>medley>sources>PRETTY.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
::
:: Copyright (c) 1984-1990, 1999, 2018, 2023 by Venue & Xerox Corporation.
:: The following program was created in 1984 but has not been published
:: within the meaning of the copyright law, is furnished under license,
:: and may not be used, copied and/or disclosed except in accordance
:: with the terms of said license.
```

(RPAQQ PRETTYCOMS

```
[ (FNS PRETTYDEF PRETTYDEFCOMS PRETTYDEF0 PRETTYDEF1 PRINTDATE PRINTDATE1 PRINTFNS PRETTYCOM PRETTYVAR
  PRETTYVARI PRETTYCOM1 ENDFILE MAKEDEFLIST PP PP* PPT PRETTYPRINT PRETTYPRINT1 PRETTYPRINT2
  PRETTYPRINT3 PRINTDEF1 SUPERPRINTEQ SUPERPRINTGETPROP CHANGEFONT)
  (FNS READARRAY PRINTARRAY READARRAY-FROM-LIST PRINTARRAY-TO-LIST)
  (COMS (DECLARE%: DONTCOPY (MACROS CHANGFONT)))
  (COMS ;COPYRIGHT
    (FNS PRINTCOPYRIGHT PRINTCOPYRIGHT1 SAVECOPYRIGHT)
    (BLOCKS (NIL PRINTCOPYRIGHT PRINTCOPYRIGHT1 SAVECOPYRIGHT (LOCALVARS . T)
      (NOLINKFNS PRINTCOPYRIGHT1)))
    (GLOBALVARS COPYRIGHTFLG COPYRIGHTOWNERS DEFAULTCOPYRIGHTKEYLST DEFAULTCOPYRIGHTOWNER
      COPYRIGHTSRESERVED)
    (INITVARS (COPYRIGHTFLG)
      (DEFAULTCOPYRIGHTOWNER)
      (COPYRIGHTPRETTYFLG T)
      (COPYRIGHTOWNERS)
      [DEFAULTCOPYRIGHTKEYLST ' ((NONE "
        " EXPLAINSTRING "NONE - No copyright ever on this file"
        CONFIRM T RETURN 'NONE)
      [%[ "owner: " EXPLAINSTRING "[ - new copyright owner -- type one
        line of text" NOECHOFLG T KEYLST ((Y "
          " RETURN
          (SUBSTRING (CADR ANSWER)
            2 -2]
      [%] "No copyright notice now
        " EXPLAINSTRING "]" - no copyright notice now" NOECHOFLG T
        RETURN NIL]
      (COPYRIGHTSRESERVED T)
      (*NEW-INTERLISP-MAKEFILE-ENVIRONMENT* ' (:READTABLE "INTERLISP" :PACKAGE "INTERLISP" :FORMAT
        :XCCS))
      (*DEFAULT-MAKEFILE-ENVIRONMENT*))
      (GLOBALVARS COPYRIGHTOWNERS DEFAULTCOPYRIGHTKEYLST COPYRIGHTPRETTYFLG COMMENTFLG
        *DEFAULT-MAKEFILE-ENVIRONMENT* *NEW-INTERLISP-MAKEFILE-ENVIRONMENT*))
      (INITVARS (COMMENTFLG '*')
        (**COMMENT**FLG ' " **COMMENT** ")
        (PRETTYFLG T)
        (%#RPARS 4)
        (CLISPIFYPRETTYFLG)
        (PRETTYTRANFLG)
        (FONTCHANGEFLG)
        (CHANGECHARTABSTR)
        (PRETTYTABFLG T)
        (DECLARETAGSLST ' (COMPILEVARS COPY COPYWHEN DOCOPY DOEVAL@COMPILE DOEVAL@LOAD DONTCOPY
          DONTEVAL@COMPILE DONTEVAL@LOAD EVAL@COMPILE EVAL@COMPILEWHEN EVAL@LOAD
          EVAL@LOADWHEN FIRST NOTFIRST))
        (AVERAGEVARLENGTH 4)
        (AVERAGEFNLENGTH 5)
        (%#CAREFULCOLUMNS 0)
        (CHANGECHAR '%|)
        (ENDLINEUSERFN))
      (INITVARS (PRETTYDEFMACROS)
        (PRETTYPRINTMACROS)
        (PRETTYEQUIVLST)
        (PRETTYPRINTYPEMACROS)
        (FILEPKGCOMSPLST ' (DECLARE%: SPECVARS LOCALVARS GLOBALVARS PROP IFFPROP P VARS INITVARS ADDVARS
          APPENDVARS FNS ARRAY E COMS ORIGINAL BLOCKS *))
        (SYSPROPS ' (PROPTYPE ALISTTYPE DELDEF EDITDEF PUTDEF GETDEF WHENCHANGED NOTICEFN NEWCOMFN
          PRETTYTYPE DELFROMPRETTYCOM ADTOPRETTYCOM ACCESSFN ACS AMAC ARGNAMES
          BLKLIBRARYDEF BROADSCOPE CLISPCLASS CLISPCLASSDEF CLISPFORM CLISPIFYISPROP
          CLISPINFIX CLISPISFORM CLISPISPROP CLISPNEG CLISPSTYPE CLISPWORD CLMAPS CODE
          CONVERT COREVAL CROPS CTYPE EDIT-SAVE EXPR FILE FILECHANGES FILEDATES FILEDEF
          FILEGROUP FILEHISTORY FILEMAP FILETYPE GLOBALVAR HISTORY I.S.OPR I.S.TYPE INFO
          LASTVALUE LISPFN MACRO MAKE NAMESCHANGED NARGS OLDVALUE OPD SETFN SUBR UBOX
          UNARYOP VALUE \DEF CLISPBRACKET TRYHARDER]
        (BLOCKS (PRETTYPRINTBLOCK PRETTYPRINT PRETTYPRINT1 PRETTYPRINT2 (ENTRIES PRETTYPRINT)
```

```

(SPECVARS FNSLST FILEFLG))
(GLOBALVARS DECLARETAGSLST LISPXPRINTFLG SYSPROPS FILEPKGCOMSPLST DWIMLOADFNSFLG PRETTYHEADER FILERDTBL
PRETTYEQUIVLST PRETTYTRANFLG CLISPIFYPRETTYFLG LISPXHISTORY DWIMFLG USERWORDS COMMENTFLG)
[DECLARE%: EVAL@COMPILE DOCOPY (P (CL:PROCLAIM ' (CL:SPECIAL DEFAULTFONT LAMBDAFONT PRETTYCOMFONT
COMMENTFONT **COMMENT**FLG PRETTYPRINTMACROS]
(DECLARE%: DOEVAL@COMPILE DONTCOPY ; IMPORT because FILEPKG has records EXPORTed but is not
; a member of EXPORTFILES
(FILE (IMPORT)
FILEPKG))
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVERS (ADDVARS (NLAMA PPT PP* PP)
(NLAML)
(LAMA])

```

(DEFINEQ

(PRETTYDEF

```

[LAMBDA (PRTTYFNS PRTTYFILE PRTTYCOMS REPRINTFNS SOURCEFILE CHANGES FORMAT)
; Edited 9-Jul-2021 14:12 by rmk:
; Edited 29-Jun-2021 17:23 by rmk:
(DECLARE (SPECVARS PRTTYFILE REPRINTFNS SOURCEFILE CHANGES))
(RESETLST
[RESETSAVE (RESETUNDO)
' (AND RESETSTATE (RESETUNDO OLDVALUE)
;; Says undo everything if there is an error or control-D This is particularly necessary if user is using the PRINT* prettyprintmacro which
;; updates comments to point to the newest version.
(PROG ((NEWFILEMAP (AND BUILDMAPFLG (LIST NIL)))
(%#RPARS %#RPARS)
(*PRINT-ARRAY* T)
(XCL:*PRINT-STRUCTURE* T)
(*PRINT-LEVEL* NIL)
(*PRINT-LENGTH* NIL)
FILEFLG FNSLST PRTTYTEM PRETTYCOMSLST PRTTYSPELLFLG OLDFILEMAP MAPADR NLAMALST NLAMLST LAMALST
LAM?LST FILEDATES ORIGFLG ROOTNAME DESTINATIONENV SOURCEFILEENV SOURCEFC FCLOCATION DEFAULTDESTENV
)
(DECLARE (SPECVARS *PRINT-ARRAY* XCL:*PRINT-STRUCTURE* *PRINT-LEVEL* *PRINT-LENGTH* NEWFILEMAP
ORIGFLG FILEFLG NLAMALST PRTTYSPELLFLG PRETTYCOMSLST PRTTYCOMS LAM?LST FNSLST
OLDFILEMAP LAMALST MAPADR ORIGFLG NLAMLST DESTINATIONENV SOURCEFILEENV %#RPARS))
; NEWFILEMAP corresponds to the map being built for the file
; being written. OLDFILEMAP corresponds to the map that
; exists for SOURCEFILE, if any.
(COND
((OR (NULL (\DTEST PRTTYFILE 'LITATOM))
(EQ PRTTYFILE T))
; we no longer support any of the crufty alternatives to writing a
; brand new file
(\ILLEGAL.ARG PRTTYFILE)))
(SETQ ROOTNAME (ROOTFILENAME PRTTYFILE))
(if (OR (EQ SOURCEFILE T)
(AND REPRINTFNS (NULL SOURCEFILE)))
then
;; SOURCEFILE plays the role of CFILE for recompiling. It permits PRETTYPRINT to obtain the definitions from the file
;; without having to reprettyprint them, or even having them loaded into core. T (or NIL if REPRINTFNS is specified) is the
;; same as PRETTYFILE.
;; REPRINTFNS specifies those functions to be printed anew. REPRINTFNS=T means reprint all EXPRS, a la recompile.
;; For example, if you have an entire file loaded in, but only change a few functions, using this option can speed up
;; dumping the file by a factor of two. If REPRINTFNS=ALL, all functions that contain in core exprs, whether on function
;; definition cell or property lists, are reprinted. REPRINTFNS can also be a list. MAKEFILE uses this for the REMAKE
;; option by specifying as REPRINTFNS the list CHANGES. In any case, if the function does not contain an in core
;; definition, prettyprint will try to find one on the file. i.e., act as though REPRINTFNS were NIL.
(SETQ SOURCEFILE ROOTNAME))
[if (SETQ DESTINATIONENV (GET ROOTNAME 'MAKEFILE-ENVIRONMENT))
then (SETQ DESTINATIONENV (\DO-DEFINE-FILE-INFO NIL DESTINATIONENV))
else
; see if we already know the environment of the source
(CL:MULTIPLE-VALUE-SETQ (SOURCEFILEENV OLDFILEMAP SOURCEFC)
(LOOKUP-ENVIRONMENT-AND-FILEMAP (OR SOURCEFILE ROOTNAME)
(OR (NULL SOURCEFILE)
(EQ SOURCEFILE ROOTNAME))
(if SOURCEFILE
then (if [NULL (NLSETQ (SETQ SOURCEFILE (OPENSTREAM SOURCEFILE 'INPUT])
then
; can't find file to reprint from.
; OPENSTREAM was called in order that 'correction' take place.
(SETQ SOURCEFILE NIL)
(PRINTOUT T PRTTYFILE " not found, so it will be written anew." T)
elseif (RANDACCESSP SOURCEFILE)
then (RESETSAVE NIL (LIST 'CLOSEF SOURCEFILE))
; The typical case: File is open
(RESETSAVE (INPUT SOURCEFILE))
[if (EQ REPRINTFNS 'EXPRS)
then (SETQ REPRINTFNS T)
elseif (EQ REPRINTFNS 'CHANGES)
then (SETQ REPRINTFNS (UNION (FILEPKG.CHANGEDFNS CHANGES)
(FILEPKG.CHANGEDFNS (fetch FILECHANGES
of ROOTNAME))
(CL:UNLESS SOURCEFILEENV
; if we didn't have environment cached, look it up from the actual
; stream now
(CL:MULTIPLE-VALUE-SETQ (SOURCEFILEENV OLDFILEMAP SOURCEFC)
(GET-ENVIRONMENT-AND-FILEMAP SOURCEFILE)))

```

```

(\EXTERNALFORMAT SOURCEFILE (FETCH (READER-ENVIRONMENT REFORMAT) OF SOURCEFILENV
))
(if (NULL OLDFILEMAP)
  then ; no map on file, so we will build one as needed
    (SETFILEPTR SOURCEFILE (OR SOURCEFC 0))
  elseif (NULL (CAR OLDFILEMAP))
  then ; complete map.
  elseif (LISTP (CAR OLDFILEMAP))
  then ; only partial map built up. should only happen for files that were
    ; made with BUILDMAPFLG=NIL, since otherwise there would
    ; be a complete map on the file.
    (SETFILEPTR SOURCEFILE (CAAR OLDFILEMAP))
  else ; Redundancy check. Should only occur if there was a compiled
    ; function in the file. and a partial map was formed that stopped
    ; after that function.
    (HELP))
else ; Can't copy from non-randaccessp source
  (SETQ SOURCEFILE NIL))
;; Now figure out what environment to write the new file in.
(CL:UNLESS DESTINATIONENV ; Don't yet have a destination environment
  (SETQ DESTINATIONENV (if SOURCEFILENV
    then ; use same as source
      (if (EQUAL-READER-ENVIRONMENT SOURCEFILENV
        *OLD-INTERLISP-READ-ENVIRONMENT*)
        then ; Coercing to the new style
          (\DO-DEFINE-FILE-INFO NIL
            *NEW-INTERLISP-MAKEFILE-ENVIRONMENT*)
        else ; use same env on new file as old. But copy in case something
          ; changes
            (CREATE READER-ENVIRONMENT USING SOURCEFILENV))
      elseif *DEFAULT-MAKEFILE-ENVIRONMENT*
      then ;; new file, use default
        (\DO-DEFINE-FILE-INFO NIL *DEFAULT-MAKEFILE-ENVIRONMENT*)))
  (CL:WHEN FORMAT
    (REPLACE (READER-ENVIRONMENT REFORMAT) OF DESTINATIONENV WITH FORMAT))
;; We now have a DESTINATIONENV
(if (NULL SOURCEFILE)
  then ; get rid of anything we knew about source
    (SETQ OLDFILEMAP NIL)
    (SETQ SOURCEFC NIL)
    (SETQ SOURCEFILENV NIL)
  elseif (AND SOURCEFILENV (EQUAL-READER-ENVIRONMENT SOURCEFILENV DESTINATIONENV))
  then ; source and destination compatible, so we won't need to worry
    ; about it in PRETTYPRINT1/2
    (SETQ SOURCEFILENV NIL))
;; Ready to go. Clean up by closing and deleting file if aborted.
(RESETSAVE NIL (LIST (FUNCTION PRETTYDEF0)
  (SETQ PRTTYFILE (OPENSTREAM PRTTYFILE 'OUTPUT)
  (\EXTERNALFORMAT PRTTYFILE (FETCH (READER-ENVIRONMENT REFORMAT) OF DESTINATIONENV))
  (RESETSAVE (OUTPUT PRTTYFILE))
  (PRINT-READER-ENVIRONMENT DESTINATIONENV)
  (SETQ FCLOCATION (GETFILEPTR PRTTYFILE))
  (WITH-READER-ENVIRONMENT DESTINATIONENV
    (CL:UNLESS (SYNTAXP (CHARCODE "["
      'LEFTBRACKET)
      (SETQ %RPARS NIL)) ; can't use brackets on this read table
      (SETQ FILEDATES (PRINTDATE PRTTYFILE CHANGES))
      (AND (NEQ COPYRIGHTFLG 'NEVER)
        ROOTNAME
        (PRINTCOPYRIGHT ROOTNAME))
      (SETQ FILEFLG T)
      (SETQ CHANGES (FILEPKG.CHANGEDFNS CHANGES)) ; Used freely by PRETTYPRINT to decide clispifying.
      (CL:UNLESS (RANDACCESSP PRTTYFILE) ; No point building a map, since we won't be able to go back to
        ; the start to point at it
        (SETQ NEWFILEMAP NIL))
      (CL:WHEN FONTCHANGEFLG ; this is expensive in that it costs as many conses as there are
        ; functions, but you can afford it for a makefile.
        (SETQ FNSLST (OR (for FL in (GETPROP ROOTNAME 'FILEGROUP)
          when (fetch FILEPROP of FL) join (FILEFNSLST FL))
            (FILEFNSLST ROOTNAME))))
      (CL:WHEN (OR (LISTP PRTTYFNS)
        (LISTP (GETTOPVAL PRTTYFNS))) ; Ancient cruft from before the days of MAKEFILE.
        (PRINTFNS PRTTYFNS T)
        (PRETTYCOM PRTTYFNS T))
      (CL:WHEN [SETQ PRETTYCOMSLST (OR (LISTP PRTTYCOMS)
        (AND (LITATOM PRTTYCOMS)
          (LISTP (GETTOPVAL PRTTYCOMS)
            (PRETTYCOM PRTTYCOMS T) ; PRETTYCOMS is just like the argument to a COMS command.
            ; see comment in prettycom1
            (for L on PRETTYCOMSLST do (PRETTYCOM (CAR L)
              NIL L)) ; The original value of PRTTYCOMS is saved so that it can be
                ; rewritten if a spelling correction occurs. The list
                ; PRETTYCOMSLST is searched by PRETTYCOM1 for *
                ; commands to see if the variable has be dumped out as well.

```

```

)
(if (PRETTYDEF1)
  then
    ; The coms were reprinted by PRETTYDEF1 due to a change to
    ; nlama and or nlaml

  elseif PRTTYSPELLFLG
  then
    ; A correction on prettycoms was performed, so dump it out aain
    ; to get the corrected version on the file.

    (PRETTYCOM PRTTYCOMS T)
    (CL:UNLESS (EQ COPYRIGHTFLG 'NEVER)
      (SAVECOPYRIGHT ROOTNAME))
    (CL:WHEN NEWFILEMAP
      (PRIN1 " ")
      (PRIN2 'DECLARE%:)
      (SPACES 1)
      (PRIN2 'DONTCOPY)
      (TERPRI)
      (SPACES 2)
      (for ADR in MAPADR do (SETQ PRTTYTEM (GETFILEPTR PRTTYFILE))
        (SETFILEPTR PRTTYFILE ADR)
        ; Write the current file positon into the filecreated expression, and
        ; then restores the file pointer.

        (PRIN2 PRTTYTEM)
        (SETFILEPTR PRTTYFILE PRTTYTEM))
      (PRIN2 (LIST 'FILEMAP NEWFILEMAP))
      ; printed instead of prettyprinted, so wont take up two pages of
      ; listing.

      (PRIN1 " ")
      (TERPRI)

      ;; Also save map, so can be used for subsequent makefiles.

      (PUTFILEMAP (FULLNAME PRTTYFILE)
        NEWFILEMAP NIL DESTINATIONENV NIL FCLOCATION))
    (ENDFILE PRTTYFILE)
    (CL:WHEN (AND FILEDATES ROOTNAME)
      (/replace FILEDATES of ROOTNAME with FILEDATES))
    (RETURN (FULLNAME PRTTYFILE))))

```

(PRETTYDEFCOMS

```

[LAMBDA (PRTTYCOMS FNSLST)
  (DECLARE (SPECVARS FNSLST)
    (PROG ((%#RPARS %#RPARS)
      (*PRINT-ARRAY* T)
      (XCL:*PRINT-STRUCTURE* T)
      (*PRINT-LEVEL* NIL)
      (*PRINT-LENGTH* NIL)
      BUILDMAPFLG PRTTYSPELLFLG ORIGFLG SOURCEFILE)
    (DECLARE (SPECVARS *PRINT-ARRAY* XCL:*PRINT-STRUCTURE* *PRINT-LEVEL* *PRINT-LENGTH* BUILDMAPFLG
      NEWFILEMAP ORIGFLG PRTTYSPELLFLG LAM?LST ORIGFLG SOURCEFILE %#RPARS))
    (if (NOT (SYNTAXP (CHARCODE "[" 'LEFTBRACKET)))
      then
        ; can't use brackets on this read table
        (SETQ %#RPARS NIL))
      (for L on [OR (LISTP PRTTYCOMS)
        (AND (LITATOM PRTTYCOMS)
          (LISTP (GETTOPVAL PRTTYCOMS))]
        do (PRETTYCOM (CAR L)
          NIL L]))

```

; Edited 19-Aug-88 16:07 by raf

(PRETTYDEF0

```

[LAMBDA (MADEFIL)
  ;; Cleans up after prettydef in case of control-d.
  (COND
    ((OPENP MADEFIL 'OUTPUT)
      (DELFILE (CLOSEF MADEFIL]))

```

(* bvm%: " 2-Aug-86 16:24")

(PRETTYDEF1

```

[LAMBDA NIL
  (PROG (PRTTYCOM PRTTYTEM PRTTYNEW)
    (COND
      [[NULL (SOME PRETTYCOMSLST (FUNCTION (LAMBDA (X)
        (AND (EQ (CAR X)
          'DECLARE%:)
          (SETQ PRTTYTEM (MEMB 'COMPILEVARS (SETQ PRTTYCOM X)))
          (EQ (CAAR (SETQ PRTTYTEM (CDR PRTTYTEM)))
            'ADDVARS]
        (AND (NULL NLAMALST)
          (NULL NLAMLST)
          (NULL LAMALST)
          (RETURN NIL))

```

(* wt%: " 9-SEP-78 16:05")

; Updates the DECLARE: for NLAMA/NLAML

;; If thee is no DECLARE: and no nlambdas, dont bother to add any. note tha if thee is IS a DECLARE:, then we must check even if
;; there are no nlambdas, because consider what happens when user changes the only lambda to a lambda

```

;; must replace the declare: by a nop addvars.
[SETQ PRTTYCOM (SUBPAIR ' (NLAMALST NLAMLST LAMALST)
  (LIST NLAMALST NLAMLST LAMALST)
  ' (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
    (ADDVARS (NLAMA . NLAMALST)
              (NLAML . NLAMLST)
              (LAMA . LAMALST)
            )
  )
(COND
  ((AND (LISTP PRETTYCOMSLST)
        (NLISTP PRTTYCOMS))
    (/NCONC1 PRETTYCOMSLST PRTTYCOM)
    (PRETTYCOM PRTTYCOMS T)
  )
  ([NOT (EQUAL (CAR PRTTYTEM)
                (SETQ PRTTYNEW (LIST 'ADDVARS [CONS 'NLAMA (UNION NLAMALST (INTERSECTION LAM?LST
                                                                 (CDADAR PRTTYTEM)
                                                                 [CONS 'NLAML (UNION NLAMLST (INTERSECTION LAM?LST
                                                                 (CDR (CADDAR PRTTYTEM)
                                                                 (CONS 'LAMA (UNION LAMALST (INTERSECTION
                                                                 LAM?LST
                                                                 (CDR (CADDR (CAR PRTTYTEM)
                                                                 ; The reason for the unions and intersections is that prettydef simply may not know the fntyps of some of the functions in the file,
                                                                 ; namely those on lam?lst, and therefore tese should not be removed from NLAMA and NLAML if they are there from a previous
                                                                 ; makefile.
                                                                 (/RPLACA PRTTYTEM PRTTYNEW)
                                                                 (AND (NLISTP PRTTYCOMS)
                                                                 (PRETTYCOM PRTTYCOMS T)))
                                                                 (T (RETURN NIL)))
    (PRETTYCOM PRTTYCOM)
    (RETURN T])

```

(PRINTDATE

```

[LAMBDA (OUTSTREAM CHANGES) (* bvm%: " 1-Aug-86 15:51")

```

;;; assumes that OUTSTREAM is a file open for output, and prints the date information for that file on it

```

(PROG ((DAT (DATE))
      (ROOTNAME (ROOTFILENAME OUTSTREAM))
      (PREVPAIR FILEDATES)
      (if FILEPKGFLG
          then [if ROOTNAME
                then (/replace FILECHANGES of ROOTNAME with (SETQ CHANGES (FILEPKG.MERGECHANGES CHANGES
                                                                 (fetch FILECHANGES
                                                                 of ROOTNAME)
                                                                 ; The reason for the order of arguments in UNION is so that the changes will be listed in roughly the order made.
                                                                 [SETQ FILEDATES (CONS (create FILEDATEPAIR
                                                                 FILEDATE _ DAT
                                                                 DATEFILENAME _ (FULLNAME OUTSTREAM))
                                                                 (SETQ PREVPAIR (LAST (fetch FILEDATES of ROOTNAME)

```

;;; Right now, FILEDATES simply keeps latest version and date, and original version and date. Latter for integrity checks on makefile remake, as described in filepackage. Note that don't want to change FILEDATES property until PRETTDEF completes. In case of control-d, the file will be deleted automatically.

```

)
(PRINTDATE1 OUTSTREAM CHANGES DAT (fetch FILEDATE of (CAR PREVPAIR))
  (fetch DATEFILENAME of (CAR PREVPAIR))) ; PRINTDATE1 does the actual printing. It is a separate function
; so that it can be advised.
(RETURN FILEDATES])

```

(PRINTDATE1

```

[LAMBDA (OUTSTREAM CHANGES DAT PREVDATE PREVERS TERMINATING.STRING)
; Edited 8-Feb-2023 16:21 by lmm
; Edited 30-Nov-2021 21:31 by larry
(* bvm%: "18-Sep-86 19:08")

```

;;; does the printing for PRINTDATE

```

(printout OUTSTREAM .FONT DEFAULTFONT "(" .P2 'FILECREATED %, .P2 DAT %, .FONT LAMBDAFONT .P2 (FULLNAME
OUTSTREAM)
.FONT DEFAULTFONT)
;; note that CHANGEFONT checks for FONTCHANGEFLG explicitly so that it won't do anything if FONTCHANGEFLG is NIL
(if (AND BUILDMAPFLG (NOT (DISPLAYP OUTSTREAM)))
  then (push MAPADR (ADD1 (GETFILEPTR OUTSTREAM)))
      (PRIN3 " " " OUTSTREAM)
  ; The address of where the map begins will be stored in this slot. 8 spaces left because when radix is 8, can overflow seven spaces by
  ; a file of 300000 characters (Alice did it). The push is because of a feature no longer used where there could be two FILECREATED
  ; expressions at the head of a file font
  )
[if FILEPKGFLG
  then (if INITIALS
        then (printout OUTSTREAM T T 6 .P2 :EDIT-BY %, .P2 INITIALS))

```

```

(if CHANGES
  then (printout OUTSTREAM T T 6 .P2 :CHANGES-TO %, .PPVTL CHANGES))
(if PREVDATE
  then (printout OUTSTREAM T T 6 .P2 :PREVIOUS-DATE %, .P2 PREVDATE)
  (if PREVERS
    then (printout OUTSTREAM %, .P2 PREVERS]
(PRIN1 (OR TERMINATING.STRING "
)
OUTSTREAM])

```

(PRINTFNS

```

[LAMBDA (X PRETTYDEFLG)
  (AND X (PROG (FNADRLST)
    [COND
      ((AND PRETTYDEFLG NEWFILEMAP)
        (SETQ FNADRLST (TCONC NIL (GETFILEPTR PRTTYFILE)))
        (TCONC FNADRLST NIL)
        (NCONC1 NEWFILEMAP (CAR FNADRLST]
      (PRIN1 '%()
      (PRINT 'DEFINEQ)
      (PRETTYPRINT X (AND PRETTYDEFLG (OR FNADRLST T))
        FNSLST)
      (PRIN1 '%))
      (AND FNADRLST (RPLACA (CDAR FNADRLST)
        (GETFILEPTR PRTTYFILE)))
      (TERPRI])

```

(* Imm "13-OCT-82 16:44")
; prettydeflg=T when called from prettydef.
; FNSLST bound in prettydef to list of functions on this file. used
; for font stuff.

(PRETTYCOM

```

[LAMBDA (PRTTYCOM PRTTYFLG PRETTYCOMSTAIL)
  (PROG (PRTTYTEM)
    [COND
      ((NULL PRTTYCOM)
      (RETURN))
      ((AND PRTTYFLG (NEQ PRTTYFILE T))
      (PRINT (COND
        (LISPXPRTINFLG
          ;; PRETTYCOMPRINT is an nlambda that does a lispxprint, except when prettyheader is NIL, in high case it
          ;; does nothing.
          (LIST 'PRETTYCOMPRINT PRTTYCOM))
        (T (LIST 'PRINT (LIST 'QUOTE PRTTYCOM)
          T T])
      (COND
        ((LITATOM PRTTYCOM)
        (COND
          ((AND (NULL PRTTYFLG)
            (NOT (BOUNDP PRTTYCOM))
            DWIMFLG
            (SETQ PRTTYTEM (FIXSPELL PRTTYCOM 70 USERWORDS T PRETTYCOMSTAIL (FUNCTION BOUNDP)))
            (SETQ PRTTYSPELLFLG T))
          (SETQ PRTTYCOM PRTTYTEM)))
        (PRETTYVAR PRTTYCOM PRTTYFLG)
        ;; FNS and VARS are printed as (RPAQQ atom value T) so that LOAD ALLPROP will still stre them in the value cell.
        (RETURN PRTTYCOM))
        (PRTTYFLG
          (RETURN PRTTYCOM)))
      (RETURN PRTTYCOM)))
TOP [COND
  [[AND (NULL ORIGFLG)
    (SETQ PRTTYTEM (fetch (FILEPKGCOM MACRO) of (CAR PRTTYCOM]
    (for X on (SUBPAIR (CAR PRTTYTEM)
      (PRETTYCOM1 PRTTYCOM T T)
      (CDR PRTTYTEM))
    do (PRETTYCOM (CAR X)
      NIL
      (AND PRETTYCOMSTAIL X]
  (T (SELECTQ (CAR PRTTYCOM)
    (FNS (PROG (PRTTYSPELLFLG)
      (PRINTFNS (PRETTYCOM1 PRTTYCOM T T)
        (NOT (NULL PRETTYCOMSTAIL)))
      (AND PRTTYSPELLFLG (EQ (CADR PRTTYCOM)
        '*))
      (LITATOM (SETQ PRTTYTEM (CADDR PRTTYCOM)))
      (PRETTYCOM PRTTYTEM))
      ; The FNSlst had an error in it that was corrected.
    ))
  ((VARS ARRAY)
  (for X in (PRETTYCOM1 PRTTYCOM T T) do (PRETTYVAR X)))
  (DECLARE%:
    ;; Normally, expressions appearing in a symbolic file are (1) evaluated upon loading the file, (2) not evaluated
    ;; when compiling the file, and (3) copied to the compile file. DECLARE: can be used to change state around

```

:: any PRETTYCOM. The atomic symbols DONTCOPY, DOCOPY, DONTEVAL@COMPILE,
:: DOEVAL@COMPILE, DONTEVAL@LOAD, and DOEVAL@LOAD have the obvious meaning. DECLARE:
:: eliminates the pretty commands DECLARE, COMPROP, COMPROP*, PD, PC, and PC*. DECLARE: is
:: defined as a function that evaluates all list expressions except when under a DONTEVAL@LOAD state.

```
(PRIN1 "(")
(PRIN2 'DECLARE%:)
(SPACES 1)
(for LST on (PRETTYCOM1 PRTTYCOM T T)
  do (COND
    ((NLISTP (CAR LST))
     [COND
      ((NOT (MEMB (CAR LST)
                  DECLARETAGSLST))
       (COND
        ((AND DWIMFLG (FIXSPELL (CAR LST)
                                70 DECLARETAGSLST T LST))
         (SETQ PRTTYSPELLFLG T))
         (T (GO ERROR])
        (PRIN2 (CAR LST))
        (SPACES 1))
       (T (TERPRI)
          (PRETTYCOM (CAR LST)
                     NIL LST)))
      (SELECTQ (CAR LST)
               ((EVAL@LOADWHEN EVAL@COMPILEWHEN COPYWHEN)
                (COND
                 ((SETQ LST (CDR LST))
                  (PRINTDEF (CAR LST))
                  (SPACES 1))))
               NIL))
    (PRIN1 ')
    ")
  (CL:EVAL-WHEN) ;; Has the syntax (EVAL-WHEN (times ...) coms ...). Dumps an EVAL-WHEN form on the file
                ;; containing whatever is dumped by the given COMS.
  (CL:ASSERT [AND (CL:CONSP (CADR PRTTYCOM))
                  (CL:SUBSETP (CADR PRTTYCOM)
                              '(EVAL CL:EVAL COMPILE CL:COMPILE LOAD CL:LOAD)
                              NIL "The first argument to the ~S command must be a list of times")
  (CL:FORMAT T "(~S ~S" 'CL:EVAL-WHEN (CADR PRTTYCOM))
  (for LST on (PRETTYCOM1 (CDR PRTTYCOM)
                        T NIL)
    do (CL:TERPRI)
       (PRETTYCOM (CAR LST)
                  NIL LST))
  (CL:FORMAT T "~&~%"
  ((SPECVARS LOCALVARS GLOBALVARS)
   (SETQ PRTTYTEM (CONS (CAR PRTTYCOM)
                        (PRETTYCOM1 PRTTYCOM T T)))
  (PRIN1 "(")
  (MAPRINT ' (DECLARE%: DOEVAL@COMPILE DONTCOPY)
  NIL NIL NIL NIL (FUNCTION PRIN2))
  (TERPRI)
  (PRINTDEF1 PRTTYTEM)
  (PRIN1 ")
  ")
  ((PROP IFPROP)
   [PROG ((PRTTYFLG (EQ (CAR PRTTYCOM)
                        'IFPROP))
          (PRTTYTEM (CADR PRTTYCOM))
          (PRTTYX (PRETTYCOM1 (CDR PRTTYCOM)
                              T T)))
         ; IFPROP only dumps those property values that are non-NIL.
   (COND
    ((LISTP PRTTYTEM)
     (for X in PRTTYTEM do (MAKEDEFLIST PRTTYX X PRTTYFLG)))
    ((NEQ PRTTYTEM 'ALL)
     (MAKEDEFLIST PRTTYX PRTTYTEM PRTTYFLG PRTTYCOM))
    [(ASSOC 'PUTPROPS PRETTYPRINTMACROS)
     (for ATM in PRTTYX
      do (PRINTDEF1 (CONS 'PUTPROPS
                          (CONS ATM (CONS (for X on (GETPROPLIST ATM)
                                           by (CDDR X)
                                           unless (MEMB (CAR X)
                                                         SYSPROPS)
                                           join (LIST (CAR X)
                                                         (CADR X))
                                           (T (for ATM in PRTTYX do (printout NIL %,, "(" .P2 'PUTPROPS %, .P2 ATM)
                                               (SETQ PRTTYTEM (ADD1 (POSITION)))
                                               (for X on (GETPROPLIST ATM) by (CDDR X)
                                                unless (MEMB (CAR X)
                                                            SYSPROPS)
                                                do (printout NIL .TAB PRTTYTEM .PPV (CAR X)
                                                    %, .PPV (CADR X)))
                                               (PRIN1 ')
                                               "])
      ; Arbitrary expression to evaluate when loaded. Be sure to
      ; prettyprint as code
  (P
```

```

(for X in (SETQ PRTTYTEM (PRETTYCOM1 PRTTYCOM T)) do (PRINTDEF1 X T)))
(INITVARS [for X in (PRETTYCOM1 PRTTYCOM T) do (COND
  ((LISTP X)
   (OR (EQ (CAR X)
           COMMENTFLG)
        (PRETTYVAR1 'RPAQ? (CAR X)
                     (CDR X)
                     NIL T)))
   (T (PRETTYVAR1 'RPAQ? X NIL]))
(ADDVARS (for X in (PRETTYCOM1 PRTTYCOM T T)
  do (PRETTYVAR1 'ADDDTOVAR [CAR (OR (LISTP X)
                                     (ERRORX (LIST 4 X)
                                               (CDR X)
                                               NIL T)))
(APPENDVARS (for X in (PRETTYCOM1 PRTTYCOM T T)
  do (PRETTYVAR1 'APPENDTOVAR [CAR (OR (LISTP X)
                                       (ERRORX (LIST 4 X)
                                               (CDR X)
                                               NIL T)))
(E (for X in (PRETTYCOM1 PRTTYCOM T) do (EVAL X)))
(COMS (for X on (PRETTYCOM1 PRTTYCOM T) do (PRETTYCOM (CAR X)
  NIL
  (AND PRETTYCOMSTAIL X))))
(ORIGINAL [LET ((ORIGFLG T)
  (DECLARE (SPECVARS ORIGFLG))
  (for X on (PRETTYCOM1 PRTTYCOM T) do (PRETTYCOM (CAR X)
  NIL
  (AND PRETTYCOMSTAIL X]))
(BLOCKS (SETQ PRTTYTEM (PRETTYCOM1 PRTTYCOM T T))
  (PRIN1 "(")
  (MAPRINT ' (DECLARE%: DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY)
  NIL NIL NIL NIL (FUNCTION PRIN2))
  (TERPRI)
  (for X in PRTTYTEM do (PRINTDEF1 (CONS 'BLOCK%: X)))
  (PRIN1 ")")
  (PRIN1 ")")
((*)
  [COND
    ((EQ (CADR PRTTYCOM)
         '*))
      ; Form-feed if super-comment indicated. Use * no matter what
      ; current COMMENTFLG is.
      (printout NIL .PAGE))
    (T (RPTQ 3 (TERPRI))
      (COND
        ((AND [COND
              (FILEFLG FONTCHANGEFLG)
              (T (EQ FONTCHANGEFLG 'ALL)
                 LAMBDAFONT)
              (CHANGEFONT LAMBDAFONT)
              (PRIN2 PRTTYCOM)
              (CHANGEFONT DEFAULTFONT))
          (T (PRIN2 PRTTYCOM)))
         (RPTQ 2 (TERPRI)))
      (COND
        ((AND (LITATOM (CAR PRTTYCOM))
              (fetch (FILEPKGTYPE GETDEF) of (CAR PRTTYCOM)))
          ; If its the name of a type with a GETDEF, put out PUTDEF
          ; expressions.
          (printout NIL "(" .P2 'PUTDEF %, .P2
                    (KWOTE X)
                    %, .P2 (KWOTE (CAR PRTTYCOM))
                    %, .PPV (KWOTE (GETDEF X (CAR PRTTYCOM)))
                    ")" T)))
        ((FIXSPELL (CAR PRTTYCOM)
                    70 FILEPKGCOMSPLST T PRTTYCOM)
         (SETQ PRTTYSPELLFLG T)
         (GO TOP))
        (T (GO ERROR]
      (RETURN PRTTYCOM)
ERROR
(ERROR "bad file package command" PRTTYCOM])

```

Prettyvar

```
[LAMBDA (VAR FLG)
```

(* Imm "27-Aug-84 20:15")

; I don't see what FLG is used for--rmk

```
(PROG (VAL TEM)
```

;; Dumps value of VAR for reloading. If VAR is non-atomic, of form (var form) where VAR is to be dumped so as to be set to value of form, ;; computed at LOAD time.

```

(COND
  ((LITATOM VAR)
   (AND (EQ (SETQ VAL (GETTOPVAL VAR))
            'NOBIND)
        (printout T T "****WARNING: " .P2 VAR " is unbound" T T))
  (PRETTYVAR1 'RPAQQ VAR VAL))
[ (LISTP VAR)

```



```

(SETQ VAL (CDR VAR))
(SETQ VAR (CAR VAR))
(COND
  ((AND (EQ [CAR (SETQ TEM (LISTP (CAR (LISTP VAL)
    'QUOTE)
    (LISTP (CDR TEM)))
    (PRETTYVAR1 'RPAQQ VAR (CADR TEM)))
    ((EQ VAR COMMENTFLG)
      ; don't print out comments
  )
  ([OR (NULL VAL)
    (AND (LISTP VAL)
      (OR (NUMBERP (CAR VAL))
        (EQ T (CAR VAL))
        (NULL (CAR VAL)))
      (NULL (CDR VAL)
        ; A minor optimization for RPAQQ's to suppresses unnecessary
        ; load-time eval's.
    (PRETTYVAR1 'RPAQQ VAR (CAR VAL)))
  (T (PRETTYVAR1 'RPAQ VAR VAL NIL T]
  (T (ERROR "Bad variable specifiation" VAR])

```

PrettyVar1

[LAMBDA (OP VAR E DEF TAILFLG) ; Edited 10-Feb-87 18:01 by Pavel

;; does printing for VAR, ADDVAR, and PROP commands. OP is the name of the function, VAR the operand, and E the rest of the expression to be printed, either as an element, or as a tail if TAILFLG=T. VAR is printed in LAMBDAFONT. If VAR is a list, each element is printed in LAMBDAFONT. This option is used to print both the name of the atom and its property for PROP commands.

```

(PROG ((LASTCOL (LINELENGTH))
  TEM
  (*PRINT-ARRAY* T))
  ; This is supposed to be bound above here but isn't in some case
  ; I can't find. --Pavel
  (TERPRI)
  ; because if you have a really bold font, it lines up the bottoms, but you can get crowded into the line above.

```

```

(COND
  ((AND (MEMB OP ' (RPAQQ RPAQ RPAQ?))
    (EQ [TYPENAME (SETQ TEM (COND
      (TAILFLG (CAR E))
      (T E)
    'ARRAYP))
    ; dump arrays and bitmaps specially. This really ought to be handled by having *PRINT-ARRAY* say how to dump these, so that
    ; only a single expression ends up on the file.
    ; 1 December 1986, Pavel: Well, I fixed bitmaps for this. Maybe I'll fix arrays as well...
    ; 10 February 1987, Pavel: ARRAYP's are now fixed as well, but not by using *PRINT-ARRAY*. Rather than invent another
    ; non-standard reader macro, I simply store the elements in a list and use a non-READING version of READARRAY.

```

```

(COND
  ((EQ OP 'RPAQQ)
    (SETQ OP RPAQ))
  (printout NIL "(" .P2 OP %, .P2 VAR %,
  [PRIN2 ` (READARRAY-FROM-LIST , (ARRAYSIZE TEM)
    ', (ARRAYTYP TEM)
    ', (ARRAYORIG TEM)
    ', (PRINTARRAY-TO-LIST TEM]
  (printout NIL '%) T))
  ((ASSOC OP PRETTYPRINTMACROS)
    (OR TAILFLG (SETQ E (CONS E)))
    (PRINTDEF [CONS OP (COND
      ((LISTP VAR)
        (APPEND VAR E))
      (T (CONS VAR E)
        0 DEF))
  (T (PRIN1 '%()
  (PRIN2 OP)
  (SPACES 1)
  (SETQ TEM (POSITION))
  (COND
    ((AND FONTCHANGEFLG PRETTYCOMFONT)
      (CHANGEFONT PRETTYCOMFONT))
    (COND
      ((LISTP VAR)
        (MAPRINT VAR NIL NIL NIL NIL (FUNCTION PRIN2)))
      (T (PRIN2 VAR)))
    (COND
      ((AND FONTCHANGEFLG PRETTYCOMFONT)
        (CHANGEFONT DEFAULTFONT))
      (SPACES 1)
      (PRINTDEF E (COND
        ((OR (NLISTP E)
          (FITP E NIL NIL LASTCOL))
          (POSITION))
        (T TEM))
        DEF TAILFLG)
  (PRIN1 '%)]
  (TERPRI])

```

(PRETTYCOM1

```
[LAMBDA (PRTYCOM PRTYFLG REMOVECOMMENTS)
  (PROG (PRTYX)
    [COND
      ((AND (EQ [CAR (LISTP (SETQ PRTYX (CDR PRTYCOM)
        '*)
          (CDR PRTYX))
        (COND
          ((AND (LITATOM (SETQ PRTYX (CADR PRTYX)))
            PRTYFLG)
            ; Checks to see if the variable is already being dumped and
            ; dumps it if not.
            (PRETTYCOM PRTYX)))
          (SETQ PRTYX (COND
            (PRTYFLG (EVAL PRTYX))
            ((LITATOM PRTYX)
              (AND (NEQ (SETQ PRTYX (GETTOPVAL PRTYX))
                'NOBIND)
                PRTYX))
            (T (RESETVARS (DWIMLOADFNSFLG)
              (RETURN (AND (ERSETQ (SETQ PRTYX (EVAL PRTYX)))
                PRTYX]
            (RETURN (if (AND REMOVECOMMENTS (LISTP PRTYX))
              then [SUBSET PRTYX (FUNCTION (LAMBDA (X)
                (OR (NLISTP X)
                  (NEQ (CAR X)
                    COMMENTFLG]
              else PRTYX])
    ])
```

(ENDFILE

```
[LAMBDA (FILE)
  (PRINT 'STOP FILE)
  (CLOSEF FILE)]
(* wt%: "10-SEP-78 13:54")
```

(MAKEDEFLIST

```
[LAMBDA (X PROP FLG)
  (for Z in X bind TEM do (COND
    [[AND (LITATOM Z)
      (SETQ TEM (SOME (GETPROPLIST Z)
        [FUNCTION (LAMBDA (X)
          (EQ X PROP]
          'CDDR]
      (PRETTYVAR1 'PUTPROPS (LIST Z PROP)
        (CADR TEM)
        (OR (EQ PROP 'EXPR)
          (MEMB PROP MACROPROPS]
      (NULL FLG)
      (EXEC-FORMAT "(no ~S property for ~S)~%" PROP Z])
    ])
```

(PP

```
[NLAMBDA X
  (DECLARE (LOCALVARS . T))
  (MAPC (NLAMBDA.ARGX X)
    (FUNCTION (LAMBDA (NAME)
      (for TYPE in (TYPESOF NAME NIL ' (FIELDS)
        'CURRENT)
        do (CL:FORMAT *TERMINAL-IO* "~A definition for ~S:~%" TYPE NAME)
          (SHOWDEF NAME TYPE)]
    ])
```

(PP*

```
[NLAMBDA X
  (DECLARE (LOCALVARS . T))
  ;; Prettyprints definitions to terminal with comments not suppressed.
  (LET [(**COMMENT**FLG NIL)
    (*STANDARD-OUTPUT* (GETSTREAM T 'OUTPUT)
    (DECLARE (SPECVARS **COMMENT**FLG *STANDARD-OUTPUT*))
    (PRETTYPRINT (NLAMBDA.ARGX X])
  ])
```

(PPT

```
[NLAMBDA X
  (DECLARE (LOCALVARS . T))
  ;; Prettyprints definitions to terminal with clisp translations shown.
  (LET [(**STANDARD-OUTPUT* (GETSTREAM T 'OUTPUT)
    (DECLARE (SPECVARS *STANDARD-OUTPUT*))
    (RESETVARS ((PRETTYTRANFLG T))
    (RETURN (PRETTYPRINT (NLAMBDA.ARGX X])
  ])
```

(PRETTYPRINT

```

[LAMBDA (FNS PRETTYDEFLG FNSLST) ; Edited 11-Feb-87 11:11 by bvm:
;; PRETTYDEFLG is supplied when called from PRINTFNS. it is either a paatial file map or T, so that it is also used as a flag for whether you are
;; being called from prettydef.
;; Note that prettyprint does all of its printing to standard output file and using current readtable. it assumes that higher functions have set these
;; appropriately, as is the case when called from prettydef, pp, pp*,
(RESETLST
 [RESETSAVE NIL (LIST (FUNCTION DSPFONT)
                      (DSPFONT)
                      (GETSTREAM NIL 'OUTPUT])
 (PROG ((CLK (CLOCK 0))
        (NEWADRLST (LISTP PRETTYDEFLG))
        [FILEFLG (NOT (DISPLAYP (OUTPUT)
                                FN DEF ADR LST SKIPPEDLST TEM) ; NEWADRLST Corresponds to the current entry on
                                                                ; NEWFILEMAP. Is in TCONC format.

        [COND
          ((ATOM (SETQ LST FNS))
           (SETQ LST (EVALV FNS]
 LP (COND
     ((NLISTP LST)
      (RETURN FNS))
     ((AND FILEFLG (IGREATERP (CLOCKDIFFERENCE CLK)
                              30000)) ; Every 30 seconds say what function we're working on
      (SETQ CLK (CLOCK 0))
      (PRIN2 (CAR LST)
              T T)
      (PRIN1 ' ", " T)))
     (SETQ FN (CAR LST))
     (TERPRI) ; The initial TERPRI is not in map
     [AND NEWADRLST (TCONC NEWADRLST (LIST FN (GETFILEPTR PRTTYFILE)
                                           ; Address of start.

 LP1 (SETQ DEF (VIRGINFN FN))
     (AND PRETTYDEFLG (SELECTQ (ARGTYPE DEF)
                               (1 (SETQ NLAMLST (CONS FN NLAMLST)))
                               (2 (SETQ LAMALST (CONS FN LAMALST)))
                               (3 (SETQ NLAMALST (CONS FN NLAMALST)))
                               (NIL (SETQ LAM?LST (CONS FN LAM?LST))))
                               ; So prettydef can add the appropriate DECLARE:

 [COND
  [(NULL DEF)
   (COND
    ((AND (NULL PRETTYDEFLG)
          FN
          (BOUNDP FN)) ; No fn definition, but is a variable. Only make this check when
                      ; called via PP or PP*

    (PRINTDEF (EVALV FN)
              2))
    (T (GO NOPRINT))
    ((NULL (EXPRP DEF))
     (GO NOPRINT))
    (T (AND ADDSPELLFLG (ADDSPELL FN))
     (COND
      ((AND PRETTYDEFLG SOURCEFILE (NULL SOURCEFILEENV)
            [NULL (SELECTQ REPRINTFNS
                          (ALL T)
                          ((T EXPRS)
                           (EXPRP FN))
                          (AND (LISTP REPRINTFNS)
                               (FMEMB FN REPRINTFNS]
                          (PRETTYPRINT1 FN)) ; Was a fn to be copied from old file, and we succeeded

      )
      (T (PRETTYPRINT3 FN DEF PRETTYDEFLG) ; Prettyprint afresh

 DEFPRINTED

```

;;; At this point we have prettyprinted FN one way or another

```

(AND NEWADRLST (RPLACD (CDADR NEWADRLST)
                      (GETFILEPTR PRTTYFILE))) ; Store end address
(TERPRI) ; TERPRI is not included in map address
(SETQ LST (CDR LST))
(GO LP)
NOPRINT
(COND
 ((AND FILEFLG SOURCEFILE (PRETTYPRINT1 FN))
  (GO DEFPRINTED))
 ((AND (NULL PRETTYDEFLG)
       (SETQ TEM (EDITLOADFNS? FN))) ; only make this check when called from PP or PP*
  (LOADFNS FN TEM 'PROP)
  (COND
   ((GETPROP FN 'EXPR)
    (GO LP1)))
  (PRINT (CONS FN '(not found))
          T T))
 ((AND DWIMFLG (NULL DEF)

```

```

      (SETQ TEM (MISSPELLED? FN 70 USERWORDS (AND PRETTYDEFLG T)
                LST))
      (NEQ TEM FN))
    (/RPLACA LST (SETQ FN TEM))
    (AND NEWADRLST (FRPLACA (CADR NEWADRLST)
                           FN)) ; Fixes filemap.
    (AND PRETTYDEFLG (SETQ PRTTYSPELLFLG T))
    (GO LP1)))
  (EXEC-FORMAT "(~S not printable)~%" FN)
  (AND LISPXHISTORY (LISPXPUT '*ERROR* FN NIL (CAAR LISPXHISTORY)))
  (COND
   (NEWADRLST (SETQ TEM (NLEFT (CAR NEWADRLST)
                               2))
              (RPLACD TEM)
              (RPLACD NEWADRLST TEM)))
  LP3 (SETQ LST (CDR LST))
      (GO LP))))

```

(PRETTYPRINT1

[LAMBDA (FN)

(* bvm%: "30-Aug-86 17:25")

;;; Like BRECOMPILE1. Obtains FN from SOURCEFILE. works whether the file has previously been mapped by PRETTYDEF, LOAD, or LOADFNS
;;; (or patially mapped)

```

  (WITH-READER-ENVIRONMENT (OR SOURCEFILEENV DESTINATIONENV)
    [PROG (ADR TEM)
      (COND
       ((NULL OLDFILEMAP)
        (GO DEFQLP))
       ((PRETTYPRINT2 FN)
        (RETURN FN))
       ((NULL (CAR OLDFILEMAP))
        (RETURN NIL)) ; The entire file has been scanned.
       )
      (T (GO FNLP) ; Already inside of DEFINEQ.
         ))
    DEFQLP ; Find DEFINEQ
    (SELECTQ (SETQ TEM (RATOM SOURCEFILE))
      ((STOP NIL) ; End of file reached.
       (SETQ OLDFILEMAP (CONS NIL OLDFILEMAP)) ; Just to inform future calls to PRETTYPRINT1 not to bother ; scanning.
       (RETURN NIL))
      (% ( [COND
          ((EQ (SETQ TEM (RATOM SOURCEFILE))
              'DEFINEQ)
           (COND
            ((NULL OLDFILEMAP)
             (SETQ OLDFILEMAP (LIST T))
              ;; In case functionis found right off, OLDFILEMAP must not be left as NIL or else next call to PRETTYPRINT1 ; will not realize are already inside of DEFINEQ.
            ))
            (GO FNLP))
          (T (SKREAD SOURCEFILE '%[])
             (SKREAD SOURCEFILE TEM))
          (GO DEFQLP)
          FNLP
          (SELECTQ (SETQ TEM (RATOM SOURCEFILE))
            (% ; End of DEFINEQ.
              (GO DEFQLP))
            ((% [%[]
              NIL)
             (SCANFILEHELP))
            (SETQ ADR (SUB1 (GETFILEPTR SOURCEFILE)))
            (SETQ TEM (RATOM SOURCEFILE))
            (SETFILEPTR SOURCEFILE ADR)
            (SKREAD SOURCEFILE)
            (COND
             ((EQ TEM FN)
              (PRETTYPRINT2 FN ADR (GETFILEPTR SOURCEFILE)) ; copies the bytes.
              (RETURN FN))
             (T (SETQ OLDFILEMAP (CONS (CONS TEM (CONS ADR (GETFILEPTR SOURCEFILE)))
                                       OLDFILEMAP))
                ;; Note that this situation only occurs when (a) the entire file was not peviously scanned, e.g. if loaded with buildmapflg off, ; and (b) user is doing a remake, and (c) this functio was either dumped directly because it was changed, or else it has been ; deleted from the FNS. The function is added to OLDFILEMAP just in case it is out of order.
                (GO FNLP]))))

```

(PRETTYPRINT2

[LAMBDA (FN FROM TO)

(* bvm%: "30-Aug-86 18:13")

:: Copies function from sourcefile to prettyfile. looking it up on the map when not already given address. returns nil if not there

```
(PROG (TEM)
  (COND
    (FROM)
    ([for X in OLDFILEMAP thereis (COND
      ((NLISTP X)
        NIL)
      ((EQ (CAR X)
        FN)
        ;; occurs when remaking a file without a map, and a function is previously skipped that later
        ;; is needed.
        (SETQ TEM X))
      ((LISTP (CDDR X))
        (SETQ TEM (FASSOC FN (CDDR X]

      (SETQ FROM (CADR TEM))
      (SETQ TO (CDDR TEM)))
      (T (RETURN NIL)))
    (SETFILEPTR SOURCEFILE FROM)
    (RATOM SOURCEFILE)
```

:: The RATOM skips the paren. the reason for the RATOM instead of simply setting file ptr to (ADD1 FROM) is that there may be font info there.

```
(COND
  ((NEQ FN (SETQ TEM (READ SOURCEFILE))) ; Consistency check.
  (LISPXPRT (CONS FN TEM)
    T)
  (ERROR "filemap does not agree with contents of" SOURCEFILE T)))
(if (NULL SOURCEFILEENV)
  then ; compatible environments, just copy characters
  (COPYCHARS SOURCEFILE PRTTYFILE FROM TO)
  else ; incompatible, have to read old def and reprettyprint
  (SETQ TEM (READ SOURCEFILE)) ; old definition
  (WITH-READER-ENVIRONMENT DESTINATIONENV (PRETTYPRINT3 FN TEM T))) ; Initial and final TERPRI's are done by callers; they are not in
  ; map.

(RETURN FN))
```

(PRETTYPRINT3

[LAMBDA (FN DEF PRETTYDEFLG) (* bvm%: "30-Aug-86 17:18")

```
(LET (TEM)
  (AND (OR (SELECTQ CLISPIFYPRETTYFLG
    ((T EXPRS)
      (EXPRP FN))
    (ALL T)
    (CHANGES (AND PRETTYDEFLG (MEMB FN CHANGES)))
    (MEMB FN CLISPIFYPRETTYFLG))
    (AND (SUPERPRINTEQ (CAR (SETQ TEM (CADDR DEF)))
      COMMENTFLG)
      (EQ (CADR TEM)
        'DECLARATIONS%:)
      (MEMB 'CLISPIFY TEM)))
    (SETQ DEF (CLISPIFY DEF))))
```

:: If the function is stored on property list, only clispify if user specifically said MAKEFILE (file CLISPIFY), otherwise, assume that functions
 :: on property list have already been clispified

```
(COND
  ((AND LAMBDAFONT FONTCHANGEFLG)
    (PRIN1 ' % ())
    ;; The font change is after the paren because of problems with updating filemaps when moving back and forth between -10 and -D
    ;; systems--rmk
    (CHANGEFONT LAMBDAFONT)
    (PRIN2 FN)
    (CHANGEFONT DEFAULTFONT)
    (TERPRI))
  (T (PRIN1 ' % ())
    (PRINT FN)))
(PRINTDEF DEF 2 'FNS NIL FNSLST)
(PRIN1 ' % ())
FN])
```

(PRINTDEF1

```
[LAMBDA (EXPR FORMFLG)
  ;; Edited 19-Jan-2022 20:35 by rmk: Added DEFMACRO
  ;; Edited 16-Apr-2018 21:35 by rmk:
  ;; Edited 16-Apr-2018 10:14 by rmk:
  ;; Edited 14-Apr-88 18:21 by bvm
  ;; RMK: Special for DEFUNs: build filemap as per PRINTFNS
  ;; Used by MAKEFILE to print P, etc expressions.
  (TERPRI)
  (LET (STARTPOS ENDPOS)
```

```
(IF [AND FORMFLG NEWFILEMAP (MEMB (CAR EXPR)
                                     ' (CL:DEFUN DEFMACRO)]
    THEN (SETQ STARTPOS (GETFILEPTR PRTTYFILE))
    (PRINTDEF EXPR NIL FORMFLG NIL FNSLST)
    [IF STARTPOS
     THEN (SETQ ENDPOS (GETFILEPTR PRTTYFILE))
          (NCONC1 NEWFILEMAP (LIST STARTPOS ENDPOS (CONS (CADR EXPR)
                                                         (CONS STARTPOS ENDPOS)]
          (TERPRI])
```

(SUPERPRINTEQ

```
[LAMBDA (X Y)
 (OR (EQ X Y)
      (AND Y (EQ (CDR (FASSOC X PRETTYEQUIVLST))
                 Y]))
```

(SUPERPRINTGETPROP

```
[LAMBDA (ATM PROP)
 (OR (GETPROP (CDR (FASSOC ATM PRETTYEQUIVLST))
           PROP)
      (GETPROP ATM PROP))
 (* wt%: "17-SEP-79 15:57")
```

(CHANGEFONT

```
[LAMBDA (FONTCLASS FILE)
 (* Imm "17-Jan-86 20:59")
 ;; for calls to changefont when not under prettyprin prettydef. This is only for non-D systems. For D, DSPFONT is moved ed in.
 ;; Don't bother testing for FONTCHANGEFLG=ALL, because presumably the FONTCLASS will have a NULL entry if display printing isn't wanted.
 ;; FONTCHANGEFLG=ALL tests are really only needed if something expensive can be avoided by advance knowledge.
 (AND FONTCHANGEFLG FONTCLASS (DSPFONT FONTCLASS FILE])
```

(DEFINEQ

(READARRAY

```
[LAMBDA (SIZE TYPE ORIG)
 (* rrb " 4-JUL-80 17:07")
 ;; type is one of: POINTER, FIXP, SMALLPOSP BYTE DOUBLEPOINTER or a number which is the place (between 0 and SIZE) where FIXPs stop
 ;; and POINTERS begin.
 (PROG (X (A (ARRAY SIZE TYPE NIL ORIG))
        M DELTA)
  LP (COND
      ((NEQ (READC)
            ' %)
       (GO LP))
      (SETQ M 1)
      (SETQ DELTA (SUB1 (OR ORIG 1)))
  LP1 (COND
       ((NOT (IGREATERP M SIZE))
        (SETA A (IPLUS M DELTA)
              (READ))
        (SETQ M (ADD1 M))
        (GO LP1))
       (NULL (READ))
       ;; PRINTARRAY writes a NIL if there are no elements in the array for which the left half must be set using SETD, otherwise it writes a
       ;; T.
       (GO OUT)))
  [SETQ M (COND
          ((NUMBERP TYPE)
           (ADD1 TYPE))
          ((EQ TYPE 'DOUBLEPOINTER)
           1)
          (T (SHOULDNT])
  LP2 (COND
       ((NOT (IGREATERP M SIZE))
        (SETD A (IPLUS M DELTA)
              (READ))
        (SETQ M (ADD1 M))
        (GO LP2)))
  OUT (READ)
      ; Reads the final right parentheses surrounding the elements of
      ; the array.
  (RETURN A])
```

(PRINTARRAY

```
[LAMBDA (V)
 (* bvm%: " 3-Oct-86 12:57")
 ; Used by prettydef. Included in ABASIC because it uses LOC
 ; and VAG on the 10
 (PROG (A N M TYPE FLG DOUBLEFLG ORIG)
  [COND
   [[AND (LITATOM V)
          (ARRAYP (SETQ A (EVALV V 'PRINTARRAY))
                  (PRINT `(SETQ ,V (READARRAY , (SETQ N (ARRAYSIZE A))
```

```

                                ', (SETQ TYPE (ARRAYTYP A))
                                ', (SETQ ORIG (ARRAYORIG A)]
    ((ARRAYP V) ; Just dumps the element expression--assumes that
                                ; READARRAY has already been written
      (SETQ A V)
      (SETQ N (ARRAYSIZE A))
      (SETQ TYPE (ARRAYTYP A))
      (SETQ ORIG (ARRAYORIG A))
      (T (RETURN (HELP V "not array")])
    (PRIN1 '%())
    (SETQ DOUBLEFLG (OR (EQ TYPE 'DOUBLEPOINTER)
                        (NUMBERP TYPE))) ; note if this array has different ELTD.
    (SETQ M 1)
  LP (COND
      ((NOT (IGREATERP M N))
       (COND
         [(OR (EQ TYPE 'POINTER)
              DOUBLEFLG)
          (PRINT (ELT A (SUB1 (IPLUS M ORIG))
                  (T
                    [PRIN2 (ELT A (SUB1 (IPLUS M ORIG))
                                (SPACES 1)]
                    ; changed from PRINT to PRIN2 so would look better in file.
                    ;; check for any non-NIL entries in the ELTD part of the double arrays. If there are none, format for print out avoids lots of NILs.
                    (AND DOUBLEFLG (COND
                              ((NUMBERP TYPE)
                               (IGREATERP M TYPE)) ; check for M being in the double pointer part of the array
                              (T T))
                     (ELTD A (SUB1 (IPLUS M ORIG)))
                     (SETQ FLG T))
                    (SETQ M (ADD1 M))
                    (GO LP))
          (NULL (PRINT FLG)) ; if FLG is NULL, there are non-NIL double word entries.
          (GO OUT)))
      [SETQ M (COND
              ((EQ TYPE 'DOUBLEPOINTER)
               1) ; all entries are double
              ((NUMBERP TYPE)
               (ADD1 TYPE)) ; first TYPE elements in the array are numbers
            )
    LP1 (COND
        ((NOT (IGREATERP M N))
         [PRINT (ELTD A (SUB1 (IPLUS M ORIG))
                 (SETQ M (ADD1 M))
                 (GO LP1)))
    OUT (PRIN1 '%())
        (RETURN A)]

```

(READARRAY-FROM-LIST

[LAMBDA (SIZE TYPE ORIG ELEMENTS) ; Edited 10-Feb-87 17:59 by Pavel

;;; This is not written in the most straightforward way possible. Rather, in order to minimize the possibility of destabilization, we have kept this as much
;;; like READARRAY as possible. In essence, the only change is to use POP instead of READ.

;; type is one of: POINTER, FIXP, SMALLPOSP BYTE DOUBLEPOINTER or a number which is the place (between 0 and SIZE) where FIXPs stop
;; and POINTERs begin.

```

    (PROG (X (A (ARRAY SIZE TYPE NIL ORIG))
           M DELTA)
      LP (SETQ M 1)
         (SETQ DELTA (SUB1 (OR ORIG 1)))
      LP1 (COND
          ((NOT (IGREATERP M SIZE))
           (SETA A (IPLUS M DELTA)
                  (pop ELEMENTS))
           (SETQ M (ADD1 M))
           (GO LP1))
          ((NULL (pop ELEMENTS))
           ;; PRINTARRAY writes a NIL if there are no elements in the array for which the left half must be set using SETD, otherwise it writes a
           ;; T.
           (GO OUT)))
      [SETQ M (COND
              ((NUMBERP TYPE)
               (ADD1 TYPE))
              ((EQ TYPE 'DOUBLEPOINTER)
               1)
              (T (SHOULDNT])
    LP2 (COND
        ((NOT (IGREATERP M SIZE))
         (SETD A (IPLUS M DELTA)
                (pop ELEMENTS))
         (SETQ M (ADD1 M))
         (GO LP2)))
    OUT (RETURN A)]

```

(PRINTARRAY-TO-LIST


```

(CADR X))
' RETURN
(CADR X)
' CONFIRMFLG T]
(CONS (if (SETQ OWNER (ASSOC DEFAULTCOPYRIGHTOWNER COPYRIGHTOWNERS))
          then (LIST (CONSTANT (CHARACTER (CHARCODE LF)))
                    (CONCAT DEFAULTCOPYRIGHTOWNER "
                    ")
                    ' EXPLAINSTRING
                    (CONCAT "<LF> - " (CADR OWNER)
                              " [Default]"))
                    ' NOECHOFLG T ' RETURN (CADR OWNER))
        else ' (%
"No copyright notice now
" EXPLAINSTRING "<LF> - no copyright notice now [Default]" NOECHOFLG T RETURN NIL))
          DEFAULTCOPYRIGHTKEYLST))
          T T))
(/PUTPROP FILENAME 'COPYRIGHT (SETQ OWNER (LIST OWNER)
(COND
  ((NEQ (CAR OWNER)
        'NONE)
   (PROG ((CURRENTYEAR (SUBATOM (DATE (DATEFORMAT YEAR.LONG NO.TIME))
                                  -4 -1)))
          ;; see github Interlisp/medley issue #207 (Imm 9/11/2021)
          (OR (EQ COPYRIGHTFLG 'PRESERVE)
              (MEMBER CURRENTYEAR (CDR OWNER))
              (NCONC1 OWNER CURRENTYEAR)))
           (PRINTCOPYRIGHT1 OWNER]))

```

(PRINTCOPYRIGHT1

[LAMBDA (OWNER)

; Edited 21-Feb-2021 10:58 by rmk;
; Edited 6-Apr-90 10:36 by jds

```

(PROG ((DATES (CDR OWNER))
       (SEMICOLON (AND (READTABLEPROP *READTABLE* 'COMMONLISP)
                       "; ")))
      (PRIVATE NIL))
(COND
  ((EQ (CAR DATES)
       T)
   (SETQ PRIVATE T)
   (pop DATES)))
(COND
  (SEMICOLON
   (PRIN1 SEMICOLON))
  (T
   ; do CommonLisp style comment
   ; Print IL-style comment, with a ; in it so the pretty printer will
   ; render it as a CL-style comment.
   (printout NIL "(* ; %" T)))
  (PRIN3 "Copyright (c) ")
  [for Y START END on DATES do
   ; print years of copyright, e.g., 1985, 1986. Print intervals for
   ; successive years
   (SETQ START (SETQ END (CAR Y)))
   (FOR NEXT IN (CDR Y) WHILE (EQ (ADD1 END)
                                   NEXT)
    DO (SETQ END NEXT)
        (POP Y))
   (PRIN3 START)
   (CL:UNLESS (EQ START END)
    (PRIN3 "-")
    (PRIN3 END))
   (COND
    ((CDR Y)
     (PRIN3 ", "])
   (PRIN3 " by ")
   (PRIN3 (CAR OWNER))
   (PRIN3 ".")
   (AND COPYRIGHTSRESERVED (PRIN3 " All rights reserved.))
   (TERPRI)
  [COND
    (PRIVATE (for LINE in (CONS (CONCAT "The following program was created in " (CAR DATES)
                                     " but has not been published"
                                     ' ("within the meaning of the copyright law, is furnished under
                                     license," "and may not be used, copied and/or disclosed except in
                                     accordance" "with the terms of said license.))
    do (COND
        (SEMICOLON (PRIN1 SEMICOLON))
        (PRIN3 LINE)
        (TERPRI]
  (COND
    ((NOT SEMICOLON)
     (PRIN3 "%")
     (TERPRI)))
  (TERPRI])

```

(SAVECOPYRIGHT

```

[LAMBDA (FILENAME)
  ;; Called from PRETTYDEF to save copyright info on end of file
  (AND (NEQ COPYRIGHTFLG 'NEVER)
        (PROG (X)
              (COND
                ((SETQ X (GETPROP FILENAME 'COPYRIGHT))
                 (PRINT (LIST 'PUTPROPS FILENAME 'COPYRIGHT X)))
              )
        )
  )
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY
(BLOCK%: NIL PRINTCOPYRIGHT PRINTCOPYRIGHT1 SAVECOPYRIGHT (LOCALVARS . T)
  (NOLINKFNS PRINTCOPYRIGHT1))
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS COPYRIGHTFLG COPYRIGHTOWNERS DEFAULTCOPYRIGHTKEYLST DEFAULTCOPYRIGHTOWNER COPYRIGHTSRESERVED)
)
(RPAQ? COPYRIGHTFLG )
(RPAQ? DEFAULTCOPYRIGHTOWNER )
(RPAQ? COPYRIGHTPRETTYFLG T)
(RPAQ? COPYRIGHTOWNERS )
(RPAQ? DEFAULTCOPYRIGHTKEYLST
' ((NONE "
      " EXPLAINSTRING "NONE - No copyright ever on this file" CONFIRM T RETURN 'NONE)
  [%[ "owner: " EXPLAINSTRING "[ - new copyright owner -- type one line of text" NOECHOFLG T KEYLST
      ((Y Y "
          " RETURN (SUBSTRING (CADR ANSWER)
                               2 -2]
      [%] "No copyright notice now
          " EXPLAINSTRING "]" - no copyright notice now" NOECHOFLG T RETURN NIL)))
)
(RPAQ? COPYRIGHTSRESERVED T)
(RPAQ? *NEW-INTERLISP-MAKEFILE-ENVIRONMENT* ' (:READTABLE "INTERLISP" :PACKAGE "INTERLISP" :FORMAT :XCCS))
(RPAQ? *DEFAULT-MAKEFILE-ENVIRONMENT* )
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS COPYRIGHTOWNERS DEFAULTCOPYRIGHTKEYLST COPYRIGHTPRETTYFLG COMMENTFLG *DEFAULT-MAKEFILE-ENVIRONMENT*
  *NEW-INTERLISP-MAKEFILE-ENVIRONMENT*)
)
(RPAQ? COMMENTFLG ' *)
(RPAQ? **COMMENT**FLG ' " **COMMENT** ")
(RPAQ? PRETTYFLG T)
(RPAQ? %RPARS 4)
(RPAQ? CLISPIFYPRETTYFLG )
(RPAQ? PRETTYTRANFLG )
(RPAQ? FONTCHANGEFLG )
(RPAQ? CHANGECHARTABSTR )
(RPAQ? PRETTYTABFLG T)
(RPAQ? DECLARETAGSLST ' (COMPILEVARS COPY COPYWHEN DOCOPY DOEVAL@COMPILE DOEVAL@LOAD DONTCOPY DONTEVAL@COMPILE
  DONTEVAL@LOAD EVAL@COMPILE EVAL@COMPILEWHEN EVAL@LOAD EVAL@LOADWHEN FIRST
  NOTFIRST))
(RPAQ? AVERAGEVARLENGTH 4)
(RPAQ? AVERAGEFNLENGTH 5)
(RPAQ? %CAREFULCOLUMNS 0)
(RPAQ? CHANGECHAR ' % |)
(RPAQ? ENDLINEUSERFN )
(RPAQ? PRETTYDEFMACROS )
(RPAQ? PRETTYPRINTMACROS )

```

(RPAQ? **PRETTYEQUIVLST**)

(RPAQ? **PRETTYPRINTPEMACROS**)

(RPAQ? **FILEPKGCOMSPLST** ' (DECLARE%: SPECVARS LOCALVARS GLOBALVARS PROP IFPROP P VARS INITVARS ADDVARS APPENDVARS FNS ARRAY E COMS ORIGINAL BLOCKS *))

(RPAQ? **SYSPROPS**

' (PROPTYPE ALISTTYPE DELDEF EDITDEF PUTDEF GETDEF WHENCHANGED NOTICEFN NEWCOMFN PRETTYTYPE DELFROMPRETTYCOM ADDTOPRETTYCOM ACCESSFN ACS AMAC ARGNAMES BLKLIBRARYDEF BROADSCOPE CLISPCLASS CLISPCLASSDEF CLISPFORM CLISPIFYISPROP CLISPINFIX CLISPISFORM CLISPISPROP CLISPNEG CLISPTYPE CLISPWORD CLMAPS CODE CONVERT COREVAL CROPS CTYPE EDIT-SAVE EXPR FILE FILECHANGES FILEDATES FILEDEF FILEGROUP FILEHISTORY FILEMAP FILETYPE GLOBALVAR HISTORY I.S.OPR I.S.TYPE INFO LASTVALUE LISPFN MACRO MAKE NAMESCHANGED NARGS OLDVALUE OPD SETFN SUBR UBOX UNARYOP VALUE \DEF CLISPBRACKET TRYHARDER))

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: PRETTYPRINTBLOCK PRETTYPRINT PRETTYPRINT1 PRETTYPRINT2 (ENTRIES PRETTYPRINT) (SPECVARS FNSLST FILEFLG))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS DECLARETAGSLST LISPXPRTFLG SYSPROPS FILEPKGCOMSPLST DWIMLOADFNSFLG PRETTYHEADER FILERDTBL PRETTYEQUIVLST PRETTYTRANFLG CLISPIFYPRETTYFLG LISPXHISTORY DWIMFLG USERWORDS COMMENTFLG)

(DECLARE%: EVAL@COMPILE DOCOPY

(CL:PROCLAIM ' (CL:SPECIAL DEFAULTFONT LAMBDAFONT PRETTYCOMFONT COMMENTFONT **COMMENT**FLG PRETTYPRINTMACROS))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(FILESLOAD (IMPORT) FILEPKG)

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

(ADDTOVAR **NLAMA** PPT PP* PP)

(ADDTOVAR **NLAML**)

(ADDTOVAR **LAMA**)

(PUTPROPS **PRETTY COPYRIGHT** ("Venue & Xerox Corporation" T 1984 1985 1986 1987 1988 1989 1990 1999 2018 2023))

FUNCTION INDEX

CHANGEFONT	14	PRETTYDEF	2	PRETTYVAR	8	PRINTDEF1	13
ENDFILE	10	PRETTYDEF0	4	PRETTYVAR1	9	PRINTFNS	6
MAKEDEFLIST	10	PRETTYDEF1	4	PRINTARRAY	14	READARRAY	14
PP	10	PRETTYDEFCOMS	4	PRINTARRAY-TO-LIST	15	READARRAY-FROM-LIST	15
PP*	10	PRETTYPRINT	10	PRINTCOPYRIGHT	16	SAVECOPYRIGHT	17
PPT	10	PRETTYPRINT1	12	PRINTCOPYRIGHT1	17	SUPERPRINTEQ	14
PRETTYCOM	6	PRETTYPRINT2	12	PRINTDATE	5	SUPERPRINTGETPROP	14
PRETTYCOM1	10	PRETTYPRINT3	13	PRINTDATE1	5		

VARIABLE INDEX

%#CAREFULCOLUMNS	18	DECLARETAGSLST	18
%#RPARS	18	DEFAULTCOPYRIGHTKEYLST	18
COMMENTFLG	18	DEFAULTCOPYRIGHTOWNER	18
DEFAULT-MAKEFILE-ENVIRONMENT	18	ENDLINEUSERFN	18
NEW-INTERLISP-MAKEFILE-ENVIRONMENT	18	FILEPKGCOMSPLST	19
AVERAGEFNLENGTH	18	FONTCHANGEFLG	18
AVERAGEVARLENGTH	18	PRETTYDEFMACROS	18
CHANGECHAR	18	PRETTYEQUIVLST	19
CHANGECHARTABSTR	18	PRETTYFLG	18
CLISPIFYPRETTYFLG	18	PRETTYPRINTMACROS	18
COMMENTFLG	18	PRETTYPRINTYPEMACROS	19
COPYRIGHTFLG	18	PRETTYTABFLG	18
COPYRIGHTOWNERS	18	PRETTYTRANFLG	18
COPYRIGHTPRETTYFLG	18	SYSPROPS	19
COPYRIGHTSRESERVED	18		

MACRO INDEX

CHANGEFONT	16
------------------	----
