File created: 2-May-2022 11:38:55 **{DSK}<home>larry>medley>sources>LOADFNS.;2**

changes to: (FNS SCANFILEHELP)

previous date: 16-Apr-2018 17:38:16 {DSK}<home>larry>medley>sources>LOADFNS.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
;;
;; Copyright (c) 1983-1984, 1986-1987, 1989-1990, 2018, 2022 by Venue & Xerox Corporation.

(RPAQQ LOADFNSCOMS
       [(FNS LOADFROM LOADBLOCK GETBLOCKDEC LOADCOMP LOADCOMP? LOADVARS LOADEFS LOADFILEMAP LOADFNS
             LOADFNS-FINDFILE LOADFNS-MAKELIST)
        (FNS LOADFNSCAN SCANFILE0 SCANCOMPILEDFN SCANDEFINEQ SCANEXP SCANDECLARECOLON SCANFILE1 SCANFILE2
             TMPSUBFN RETRYSCAN SCANFILEHELP)
        (VARS (NOT-FOUNDTAG 'NOT-FOUND%:))
        (GLOBALVARS LASTWORD LOADOPTIONS SYSFILES NOT-FOUNDTAG)
        (LOCALVARS . T)
        (BLOCKS (SCANFILEBLOCK (ENTRIES LOADFNSCAN TMPSUBFN SCANFILE1)
                       LOADFNSCAN SCANFILE0 SCANCOMPILEDFN SCANDEFINEQ SCANEXP SCANDECLARECOLON SCANFILE1
                       SCANFILE2 TMPSUBFN (LOCALFREEVARS FNADRLST DICT DICT0 ADR)
                       (SPECVARS VARLST)
                       (RETFNS SCANFILE0])

(DEFINEQ

(LOADFROM
  [LAMBDA (FILE FNS LDFLG)                                          (* wt%: "21-SEP-79 12:03")
                                                                    ; 'notices' file.
    (PROG1 (LOADFNS FNS FILE LDFLG 'LOADFROM)
        [AND DWIMFLG FNS (SETQ LASTWORD (COND
                                          ((ATOM FNS)
                                           FNS)
                                          (T (CAR (LAST FNS])])

(LOADBLOCK
  [LAMBDA (FN FILE LDFLG)                                           (* bvm%: "27-Sep-86 15:17")
    (PROG (TEM)
          (OR FILE (SETQ FILE (LOADFNS-FINDFILE FN)))
          (RETURN (AND [SETQ TEM (SUBSET (OR (GETBLOCKDEC FN FILE T)
                                             (LIST FN))
                                         (FUNCTION (LAMBDA (FN)
                                                     (NOT (EXPRP (VIRGINFN FN]
                       (LOADFNS TEM FILE LDFLG])

(GETBLOCKDEC
  [LAMBDA (FN FILE FNSONLY)                                         (* bvm%: " 7-Oct-86 18:23")
    ;; Return the block declaration of FILE that contains FN.  If FNSONLY is true, returns just a list of the functions in the block.
    (OR FILE (SETQ FILE (LOADFNS-FINDFILE FN)))
    (for BLOCK in (FILECOMSLST FILE 'BLOCKS) when (MEMB FN BLOCK)
       do (RETURN (if (NULL FNSONLY)
                      then BLOCK
                    elseif (AND (CAR BLOCK)
                                (SUBSET (CDR BLOCK)
                                        (FUNCTION LITATOM)))
                    else                                            ; car of block decl is block name or NIL for no block
                         (LIST FN])

(LOADCOMP
  [LAMBDA (FILE LDFLG)                                              (* bvm%: "27-Sep-86 16:32")
    (RESETLST
        (LET ((FULLNAME (OR (FINDFILE FILE T)
                            FILE))
              BLOCKS ROOT)
             (DECLARE (SPECVARS BLOCKS))                            ; don't let block declarations get thru
             [RESETSAVE NIL (LIST [FUNCTION (LAMBDA (NAME VAL)      ; remove LOADCOMP prop if didn't finish successfully
                                              (AND RESETSTATE (PUTPROP NAME 'LOADCOMP VAL]
                                  (SETQ ROOT (NAMEFIELD FULLNAME))
                                  (GETPROP ROOT 'LOADCOMP]
             (/PUTPROP ROOT 'LOADCOMP FULLNAME)                     ; Save FULLNAME for LOADCOMP?  Do this now rather than
                                                                    ; after the LOADFNS to avoid circularity if A loadcomp's B and B
                                                                    ; loadcomp's A.
             (LOADFNS T FULLNAME LDFLG 'LOADCOMP)))])

(LOADCOMP?
  [LAMBDA (FILE LDFLG)                                              ; Edited 22-Sep-89 16:35 by bvm
```

```
    (LET* [(FOUND (FINDFILE FILE T))
           (FULLNAME (OR FOUND FILE))
           (LOADED (GETPROP (NAMEFIELD FULLNAME)
                            'LOADCOMP]
          (if [OR (NULL LOADED)
                  (AND FOUND (NOT (STRING-EQUAL LOADED FOUND]
              then ;; Do the LOADCOMP if one's never been done, or the current version is not the one that was loadcomp'ed before.  If can't find
                   ;; a current version, assume the previously  loadcomp'ed one is ok.
                   (LOADCOMP FULLNAME LDFLG))
          FULLNAME])
```

(LOADVARS

```
  [LAMBDA (VARS FILE LDFLG)
    (LOADFNS NIL FILE LDFLG VARS])
```

(LOADEFS

```
  [LAMBDA (FNS FILE)                                    (* wt%: " 9-APR-80 20:27")
    (LOADFNS FNS FILE 'GETDEF])
```

(LOADFILEMAP

```
  [LAMBDA (FILE)                                        (* wt%: "16-MAY-79 22:05")
    ;; user wants the full filemap.  scan file if necessary.  if updatemapflg=T and any changes are made, e.g. map does not exist on file, or is wrong
    ;; (due to transferring from dorado to maxc), loadfns will rewrite the map
    (LOADFNS NIL FILE NIL 'FILEMAP])
```

(LOADFNS

```
  [LAMBDA (FNS FILE LDFLG VARS)                         (* bvm%: "17-Nov-86 23:28")
```

;;; All of LOADVARS, LOADCOMP, LOADFILEMAP, LOADFROM come thru here.

```
    (DECLARE (SPECVARS FILE LDFLG VARS))                ; Used free by RETRYSCAN
    (RESETLST
        [PROG ((*PACKAGE* *INTERLISP-PACKAGE*)
               (DFNFLG DFNFLG)
               (BUILDMAPFLG BUILDMAPFLG)
               (FILEPKGFLG FILEPKGFLG)
               (ADDSPELLFLG ADDSPELLFLG)
               (LISPXHIST LISPXHIST)
               (FILECREATEDLST)
               (PRLST (AND FILEPKGFLG (FILEPKGCHANGES)))
               INSTREAM FNLST VARLST DONELST ROOTNAME FILEMAP TEM FILEMAPEND FILECREATEDLOC FILENV RESETSAVER
               MAPUPDATED)
              (DECLARE (SPECVARS *PACKAGE* DFNFLG BUILDMAPFLG FILEPKGFLG ADDSPELLFLG LISPXHIST FNLST VARLST
                               DONELST FILECREATEDLST FILECREATEDLOC))
                                                        ; FILECREATEDLST is set by SCANEXP when it encounters a
                                                        ; FILECREATED expression
        TOP (COND
               ((OR (EQ LDFLG 'EXPRESSIONS)
                    (EQ LDFLG 'GETDEF)
                    (MEMB LDFLG LOADOPTIONS))
                (SETQ DFNFLG LDFLG))
               ((AND DWIMFLG (SETQ TEM (FIXSPELL LDFLG NIL LOADOPTIONS T)))
                (SETQ LDFLG TEM)
                (SETQ DFNFLG LDFLG))
               (T (SETQ LDFLG (ERROR "unrecognized load option" LDFLG))
                  (GO TOP)))
            (COND
               ((EQ LDFLG 'SYSLOAD)
                (SETQ DFNFLG T)
                (SETQ ADDSPELLFLG NIL)
                (SETQ BUILDMAPFLG NIL)
                (SETQ FILEPKGFLG NIL)
                (SETQ LISPXHIST NIL)))
            [AND LISPXHIST (COND
                              ((SETQ TEM (FMEMB 'SIDE LISPXHIST))
                               (FRPLACA (CADR TEM)
                                        -1))
                              (T (LISPXPUT 'SIDE (LIST -1)
                                           NIL LISPXHIST]
                                                        ; So that UNDOSAVE will keep saving regardless of how many
                                                        ; undosaves are involved
            (SETQ FNLST (LOADFNS-MAKELIST FNS T))       ; Get list of functions
            [COND
               ((NULL FILE)                             ; Infer what file caller meant (this is a feature!)
                (SETQ FILE (LOADFNS-FINDFILE (CAR FNLST]
        RETRY
            [RESETSAVE NIL (SETQ RESETSAVER (LIST 'CLOSEF? (SETQ INSTREAM (OPENSTREAM FILE 'INPUT]
                                                        ; CLOSEF?  not CLOSEF because UPDATEFILEMAP might
                                                        ; close file for us
            (RESETSAVE (INPUT INSTREAM))
            (SETQ FILE (FULLNAME INSTREAM))             ; Gets full file name.  Also note that there may have been some
                                                        ; error correction done in OPENSTREAM
```

```
(COND
   ((NOT (RANDACCESSP INSTREAM))
    (SETQ FILE (ERROR FILE "not a random access file"))
    (GO RETRY)))
(SETFILEPTR INSTREAM 0)
(SETQ ROOTNAME (ROOTFILENAME FILE))
(CL:MULTIPLE-VALUE-SETQ (FILENV FILEMAP FILECREATEDLOC FILECREATEDLST)
       (GET-ENVIRONMENT-AND-FILEMAP INSTREAM))
(SETQ VARLST (SELECTQ VARS
                  (NIL NIL)
                  (VARS                                ; Means load, i.e., evaluate, ALL rpaq/rpaqq
                        'VARS)
                  (FNS/VARS (LIST (FILECOMS ROOTNAME 'COMS)
                                  (FILECOMS ROOTNAME 'BLOCKS)))
                  (LOADCOMP                            ; evaluate the EVAL@COMPILE expresions, notice the fns and
                                                       ; vars.
                            (SETQ FNLST T)
                            VARS)
                  (FILEMAP                             ; Return the filemap, or build one if not already available
                           (if (AND FILEMAP (NULL (CAR FILEMAP)))
                               then (RETURN FILEMAP)
                             elseif (NULL BUILDMAPFLG)
                               then (RETURN NIL))
                           'FILEMAP)
                  (LOADFROM ;; evaluate all non-defineq expressions, but just return file name as value, i.e. dont bother
                            ;; adding to donelst
                            'LOADFROM)
                  (DONTCOPY                            ; means load all DECLARE: DONTCOPY expressions
                            VARS)
                  (LOADFNS-MAKELIST VARS)))
(SETQ FILEMAPEND (if FILEMAP
                     then (CAR FILEMAP)
                   else T))                            ; Remember how far the filemap scan got already
(WITH-READER-ENVIRONMENT FILENV
    (SETQ FILEMAP (LOADFNSCAN FILEMAP))
```

;;; SCANFILE0 returns a 'map' for the file. The form of the map is (ADR ADRLST ADRLST ...) where ADR is last address scanned to in file, or NIL if ;;; entire file was scanned, or (ADR) where the scan stopped after a function in the middle of a DEFINEQ. Each ADRLST is either of the form (ADR1 ;;; ADR2 . FN) or (ADR1 ADR2 (FN ADRX . ADRY) (FN ADRX . ADRY) ...).

;;; The first case corresponds to a compiled function, the second to a DEFINEQ. In the first case, ADR1 is the address of the first character AFTER the ;;; function name in the file (for use by LAPRD) and ADR2 the address of the first character after the de definition, i.e., after LAPRD or LCSKIP has ;;; finished.

;;; In the second case, ADR1 is the address of the lef paren before the DEFINEQ, and ADR2 either the address of the first character after the entire ;;; DEFINEQ expression, or the address of the first chracter after the last function that was scanned. In (FN ADRX . ADRY), ADR is the address of of the ;;; left parentheses before the function name, ADRY the address of the character after the right paren that closes the definition.

;;; A map of non-functions is not kept because (a) it would not be of use to MAKEFILE since it always recomputes VARS, and (B) most requests for other ;;; than functions require scanning the entire file anyway, e.g. to find all RPAQQ's, and (C) the expressions are usually small compared to DEFINEQ's.

```
[if FILEMAP
    then
    (if (NEQ FILEMAPEND (CAR FILEMAP))
        then                                           ; something was added
             (PUTFILEMAP FILE FILEMAP FILECREATEDLST)
             (if (AND UPDATEMAPFLG (UPDATEFILEMAP INSTREAM FILEMAP))
                 then (SETQ MAPUPDATED T)))
    (if (AND DWIMFLG (NOT NOSPELLFLG)
             (LISTP FNLST))
        then                                           ; There are still FNS left that we didn't find
        (if (SETQ TEM
             (for X on FNLST
                bind [KNOWNFNS _ (for TRIPLE in (CDR FILEMAP)
                                    join               ; makes a list of functions found for use for spelling correction.
                                         (if (LISTP (SETQ TEM (CDDR TRIPLE)))
                                             then
                                                  ; This is for normal source files, where TRIPLE = (start end .
                                                  ; fnEntries)
                                                   (MAPCAR TEM (FUNCTION CAR))
                                           elseif TEM
                                             then
                                                  ; For compiled files, TRIPLE = (start end . fn)
                                                   (LIST TEM]
                when (AND (NOT (FMEMB (CAR X)
                                      KNOWNFNS))
                          (FIXSPELL (CAR X)
                                 70 KNOWNFNS NIL X))
                collect
                     ;; The FMEMB check is necessary for when VARS=DEFS, as the reason that the function was not
                     ;; removed from FNLST may have been because this was a compiled file.
                       (CAR X)))
            then (if MAPUPDATED
                     then                              ; UPDATEFILEMAP had closed the file
                          [RPLACA (CDR RESETSAVER)
```

```
                                                (SETQ INSTREAM (OPENSTREAM FILE 'INPUT]
                                            (INPUT INSTREAM))
                                    (SCANFILE1 FILEMAP TEM]
                            (if (AND NOT-FOUNDTAG (LISTP FNLST))
                                then (SETQ DONELST (CONS (CONS NOT-FOUNDTAG FNLST)
                                                         DONELST)))
                            (if
                             [AND NOT-FOUNDTAG (LISTP VARLST)
                                  (SETQ TEM
                                   (if (FNTYP VARLST)
                                       then (AND (NULL DONELST)
                                                 (LIST VARLST))
                                     else (for X in VARLST collect X
                                             unless (PROGN
                                                        ;; Reason for this is if user says LOADVARS (DEFLIST file), then DEFLIST is not removed
                                                        ;; from VARLST, since you want all such instances.
                                                    (for Y in DONELST
                                                       thereis (if (ATOM X)
                                                                   then (OR (EQ X (CAR Y))
                                                                            (EQ X (CADR Y)))
                                                                 else (EDIT4E X Y]
                                then (SETQ DONELST (CONS (CONS NOT-FOUNDTAG TEM)
                                                         DONELST)))
                            (if (EQ LDFLG 'SYSLOAD)
                                then (AND (NOT (MEMB (SETQ ROOTNAME (ROOTFILENAME FILE (CDR FILECREATEDLST)))
                                                     SYSFILES))
                                          (SETQ SYSFILES (NCONC1 SYSFILES ROOTNAME)))
                                     (SMASHFILECOMS ROOTNAME)
                              elseif FILEPKGFLG
                                then (AND (NEQ VARS 'FILEMAP)
                                          (NEQ LDFLG 'EXPRESSIONS)
                                          (NEQ LDFLG 'GETDEF)
                                          (ADDFILE FILE (SELECTQ VARS
                                                               ((T LOADFROM)
                                                                  'LOADFNS)
                                                               (LOADCOMP 'LOADCOMP)
                                                               'loadfns)
                                                        PRLST FILECREATEDLST))))
                        (RETURN (if (EQ VARS 'FILEMAP)
                                    then FILEMAP
                                  elseif (EQ VARS 'LOADFROM)
                                    then FILE
                                  else (DREVERSE DONELST]))
```

(LOADFNS-FINDFILE

```
  [LAMBDA (FN)                                                    (* bvm%: "27-Sep-86 15:03")
    ;; When LOADFNS is not given a file to load from, figure out using WHEREIS
    (LET ((DWIMFLG T)
          (FILEPKGFLG T))
         (DECLARE (SPECVARS DWIMFLG FILEPKGFLG))
         (OR (EDITLOADFNS? FN)
             (AND (EQ (NARGS 'WHEREIS)
                      4)
                  (EDITLOADFNS? FN NIL NIL T))
             (ERROR FN '"'s file not found" T])
```

(LOADFNS-MAKELIST

```
  [LAMBDA (LST FNSFLG)                                            (* bvm%: " 2-Oct-86 15:40")
    ;; Turn FNS or VARS arg to LOADFNS into an actual list of functions/variables to load, or T to load all.
    (if (EQ LST T)
        then                                                      ; Eleanor's option, load every fn found in FILE.
             T
      elseif (NULL LST)
        then NIL
      elseif (LITATOM LST)
        then (LIST LST)
      elseif (NLISTP LST)
        then (ERROR '"illegal arg" LST)
      elseif (NULL FNSFLG)
        then                                                      ; VARS arg is a list of patterns, so canonicalize them
             (for Y in LST collect (EDITFPAT Y))
      else (for F in LST when (if (LITATOM F)
                                  then T
                                else (LISPXPRIN1 '" isn't a function name -- ignored.
                                                 ")
                                     NIL)
              collect F])

)

(DEFINEQ
```

(LOADFNSCAN

```
  [LAMBDA (DICT)                                          (* wt%: " 7-DEC-79 11:57")
    (PROG (ADR)
          (SCANFILE0)
          (RETURN DICT])
```

(SCANFILE0

```
  [LAMBDA NIL                                             (* bvm%: "29-Aug-86 23:15")
    (PROG (NXT NXT1 NXT2 FNADRLST (DICT0 (CDR DICT)))
          [COND
             [(NULL DICT)
              (AND BUILDMAPFLG (SETQ DICT (LIST 0]
             (FNLST                                       ; Have some filemap, so go get functions that are on the map
                    (SCANFILE1 (CDR DICT]
          (COND
             ([AND (NULL VARLST)
                   (OR (NULL FNLST)
                       (AND DICT (NULL (CAR DICT]
              ;; Either all functions were found, or else the entire file having been scaaned, no point in scanning further
              (RETURN DICT)))
          (COND
             ((AND VARLST (NEQ VARLST 'FILEMAP))
              ;; Note that at this point there may or may not be some functions to be scanned for.  in any event, since there are VARS to be
              ;; obtained, we have to start scanning at the beginning, although DICT can be of use to save scanning of DEFINEQ's.
              (SETFILEPTR NIL (OR FILECREATEDLOC 0)))
             ((LISTP (CAR DICT))                          ; The scan stopped in the middle of a DEFINEQ.
              (SETFILEPTR NIL (SETQ ADR (CAAR DICT)))
              [AND BUILDMAPFLG (SETQ FNADRLST (LCONC NIL (CAR (LAST DICT]
              (SETQ DICT0 NIL)
              (SCANDEFINEQ T))
             (DICT                                        ; Scan stopped after a compiled function.
                   (SETFILEPTR NIL (CAR DICT))
                   (SETQ DICT0 NIL)))
      PEEKLP
          (SETQ NXT1 (SKIPSEPRCODES))
          (COND
             [(OR (SYNTAXP NXT1 'LEFTPAREN)
                  (SYNTAXP NXT1 'LEFTBRACKET))            ; Opening paren and bracket.
              (SETQ ADR (GETFILEPTR))
              (READC)                                     ; Flush the peeked-at paren.
              (SETQ NXT1 (RATOM))
              (COND
                 ((EQ NXT1 'DEFINEQ)
                  (SCANDEFINEQ))
                 (T                                       ; some functions may be inside of declare:'s so have to look at
                                                          ; each expression, even if varlst=NIL
                    (SETQ NXT2 (RATOM))                   ; Corresponds to CADR of the expression.  in the file
                    (SETFILEPTR NIL ADR)                  ; file pointer now points to just before the expression..
                    (SCANEXP NXT1 NXT2 (NEQ VARLST 'LOADCOMP]
             ((OR (EQ (SETQ NXT (READ))
                      'STOP)
                  (NULL NXT))                             ; End of file.
              (AND (CAR DICT)
                   (RPLACA DICT NIL))                     ; says scan of entire map now complete
              (RETURN))
             ((LITATOM NXT)
              (SETQ ADR (GETFILEPTR))
              (SCANCOMPILEDFN NXT)))
          (GO PEEKLP])
```

(SCANCOMPILEDFN

```
  [LAMBDA (FNAME)                                         (* wt%: " 9-APR-80 20:54")
    (PROG NIL
          [COND
             (DICT0 (AND (NOT (EQP (CAAR DICT0)
                                   ADR))
                         [NOT (SETQ DICT0 (SOME DICT0 (FUNCTION (LAMBDA (X)
                                                                  (IEQP ADR (CAR X]
                         (RETRYSCAN))
                    ;; redudnacy check the SOME is bcause of the (admittedly obsucre but actually happened) case where there are DEFINEQ's
                    ;; inside of a DECLARE:..  in this case, they would appear on the filemap, but DICT0 would not have been stepped because
                    ;; the DEFINIEQ's would not have been seen in the scan.
                    (SETFILEPTR NIL (CADAR DICT0))
                    ;; We know this function is not of interest, or it ould have been picked up in SCANFILE1.  Furthermore, we know its final
                    ;; address, so no need to LCSKIP
                    (SETQ DICT0 (CDR DICT0))
                    (RETURN T))
             (BUILDMAPFLG (NCONC1 DICT (SETQ FNADRLST (CONS (GETFILEPTR)
                                                            (CONS NIL FNAME]
          [COND
             [[AND FNLST (NEQ LDFLG 'EXPRESSIONS)
```

```
                (NEQ LDFLG 'GETDEF)
                (NEQ VARS 'LOADCOMP)
                (OR (EQ FNLST T)
                    (MEMB FNAME FNLST)
                    (SOME FNLST (FUNCTION (LAMBDA (X)
                                          (TMPSUBFN FNAME X]
          ;; We want FNAME if it is on FNLST, or a SUBFN of anything on FNLST.  or if FNLST, is T, i.e. load everything.
          (LAPRD FNAME)
          (SETQ DONELST (CONS FNAME DONELST))
          [AND FNADRLST (RPLACA (CDR FNADRLST)
                                (SETQ ADR (GETFILEPTR]
          (COND
             ((AND (NEQ FNLST T)
                   (NULL (SETQ FNLST (DREMOVE FNAME FNLST)))
                   (NULL VARLST))
              (AND DICT (RPLACA DICT ADR))
              (RETFROM 'SCANFILE0]
            (T (LCSKIP FNAME)
               (AND FNADRLST (RPLACA (CDR FNADRLST)
                                     (GETFILEPTR]
       (RETURN T])
```

(SCANDEFINEQ

```
[LAMBDA (CONTINUEFLG)                                          (* bvm%: " 7-Oct-86 18:07")
```

;; Called with file pointer just after atom DEFINEQ. DICT0, if non-NIL, is the tail of DICT that corresponds to how far we've gotten. I.e., (CAR ;; DICT0) should represent this DEFINEQ.

```
(PROG (FNAME)
      (COND
         (CONTINUEFLG (GO DEFQLP))
         ([AND DICT0 (NOT (IEQP (CAAR DICT0)
                                ADR))
               (NOT (SETQ DICT0 (find TAIL on DICT0 suchthat (IEQP ADR (CAAR TAIL]
          (RETRYSCAN)))
```

;; Double check. the SOME is because of the (admittedly obscure but it happens) case where there are DEFINEQ's inside of a DECLARE:.. in ;; this case, they would appear on the filemap, but DICT0 would not have been stepped because the DEFINEQ's would not have been seen in the ;; scan. Now we know that CAR of DICT0 corrresponds to this DEFINEQ. We process DEFINEQ's the same when there are functions to be ;; found, i.e. when FNLST is non-NIL, as when there aren't any, on the grounds that it takes about as long to do many little SKREAD's as one big ;; SKREAD, and this way we also get to build the map.

```
      [COND
         ((CADAR DICT0)
```

;; This entire DEFINEQ was scanned, and ADR is the address of the first character after it. Move file pointer and go on, i.e. dont have ;; to do SKREAD. Note that this applies even if we are looking for functions, i.e. FNLST not NIL, because in this case all functions of ;; interest would have been picked up by SCANFILE1.

```
          (SETFILEPTR NIL (CADAR DICT0))
          (SETQ DICT0 (CDR DICT0))
          (RETURN T))
         (DICT0
```

;; The scan previously stopped in the middle of a DEFINEQ. The address of the end of the scan, i.e. (CAAR DICT), ;; corresponds to the character after the last function scanned.

```
                [SETFILEPTR NIL (COND
                                   ((LISTP (CAR DICT))
                                    (CAAR DICT))
                                   (T
```

;; Another redudancy check. If the entire DEFINEQ had been processed, then CADAR of DICT0 ;; would be non-NIL, and caught above. Therefore, processing stopped in the middle of the ;; DEFINEQ, and CAR of DICT should be a list.

```
                                        (RETRYSCAN]
                [AND BUILDMAPFLG (SETQ FNADRLST (LCONC NIL (CAR DICT0]
                (SETQ DICT0 NIL))
         (BUILDMAPFLG (SETQ FNADRLST (TCONC NIL ADR))
                (TCONC FNADRLST NIL)
                (NCONC1 DICT (CAR FNADRLST]
   DEFQLP
      (SELECTQ (RATOM)
          (%)                                                  ; Closes DEFINEQ.
               (AND FNADRLST (RPLACA (CDAR FNADRLST)
                                     (GETFILEPTR)))            ; FNADRLST is a ONC format list, hence want to RPLACA
                                                               ; CDAR, not just CDR.
               (RETURN T))
          (%] (SCANFILEHELP))
          ((%( %[)
               (SETQ ADR (SUB1 (GETFILEPTR)))                  ; The address of the position of the left paren.
               (SETQ FNAME (READ))
               (AND FNADRLST (TCONC FNADRLST (LIST FNAME ADR))))
          (SCANFILEHELP))
      (SETFILEPTR NIL ADR)
```

;; Positions file pointer at left paren or bracket so if fn/def pair is closed by either right paren or bracket, read or skread will do the right thing.

```
      (COND
         [(AND FNLST (OR (EQ FNLST T)
                         (MEMB FNAME FNLST)))
```

```
            (SELECTQ VARS
                (LOADCOMP (AND (NOT (FMEMB FNAME NOFIXFNSLST))
                               (SETQ NOFIXFNSLST (CONS FNAME NOFIXFNSLST)))
                          (SKREAD))
                (SETQ DONELST (NCONC [COND
                                        ((OR (EQ LDFLG 'EXPRESSIONS)
                                             (EQ LDFLG 'GETDEF))
                                         (LIST (READ)))
                                        (T (DEFINE (LIST (READ]
                                     DONELST)))
                (AND (NEQ FNLST T)
                     (SETQ FNLST (DREMOVE FNAME FNLST]
               (T (SKREAD)))
         (AND FNADRLST (RPLACD (CDADR FNADRLST)
                               (GETFILEPTR)))
```

;; FNADRLST is a TCONC format, so its CADR is its last element. This is supposed to be of the form (FN ADRX . ADRY). This adds the ADRY.

```
         [COND
            ((AND (NULL FNLST)
                  (NULL VARLST))
```

;; Actually this check only need be made in the case that a function was actually read, i.e. second clause in above COND, but it's ;; cheap enough.

```
             [AND DICT (RPLACA DICT (LIST (ADD1 (GETFILEPTR]    ; says scan stopped in middle of defineq
             (RETFROM 'SCANFILE0]
         (GO DEFQLP])
```

(**SCANEXP**

```
[LAMBDA (EXP1 EXP2 EVALFLG)                                     ; Edited 16-Apr-2018 17:14 by rmk:
```

;; exp1 is car of the expression, exp2 cadr. file pointer is just before opening left paren and scanexp reads expression if it needs to.

```
(DECLARE (USEDFREE FILECREATEDLST))
(PROG (EXP)
      (COND
         ((EQ VARLST 'COMPILING)                                ; wants whole declare:
          (GO YES))
         ((EQ EXP1 'DECLARE%:)
          (COND
             (EXP (SETFILEPTR NIL ADR)))                        ; SKIP OVER THE PAREN AND THE DECLARE:
          (RATOM)
          (RATOM)
          (if (EQ VARLST 'DONTCOPY)
              then (SCANDECLARECOLON NIL T)
            else (SCANDECLARECOLON EVALFLG))
          (RETURN T)))
      (SELECTQ VARLST
          ((T LOADFROM)
               (AND EVALFLG (GO YES)))
          (VARS [AND EVALFLG (COND
                                 ((OR (EQ EXP1 'RPAQQ)
                                      (EQ EXP1 'RPAQ)
                                      (EQ EXP1 'RPAQ?))
                                  (GO YES])
          (LOADCOMP (AND EVALFLG (GO YES))
                    (SELECTQ EXP1
                         ((RPAQQ RPAQ RPAQ?)
                              (SETQ NOFIXVARSLST (AND (NOT (FMEMB EXP2 NOFIXVARSLST))
                                                      (CONS EXP2 NOFIXVARSLST))))
                         NIL))
          (AND (LISTP VARLST)
               [COND
                  ((FNTYP VARLST)
                   (COND
                      ((NULL (SETQ EXP (APPLY* VARLST EXP1 EXP2)))
                                                                ; the functional expression is ree to move filepinter.
                       (SETFILEPTR NIL ADR)
                       NIL)
                      ((NLISTP EXP)                             ; matched, but user elected not to return entire expression
                       (SETFILEPTR NIL ADR)
                       (SETQ EXP (READ)))
                      (T T)))
                  (T (SOME VARLST (FUNCTION (LAMBDA (X)
                                            (COND
                                               ((OR (EQ EXP1 X)
                                                    (EQ EXP2 X)))
                                               ((LISTP X)       ; edit pattern
                                                [COND
                                                   ((NULL EXP)
```

;; The expression on VARLST is a list, which is interpreted as an edit pattern; therefore we have to ;; read the entire expression from the file. Note that this is only done once, i.e., if there are several ;; patterns on VARLST, the expression from the file is read only once.

```
                                                    (SETQ EXP (READ]
                                                (EDIT4E X EXP]
               (GO YES)))
      (COND
```

```
            ((EQ EXP1 'FILECREATED)
             [SETQ FILECREATEDLST (NCONC1 FILECREATEDLST (CDR (OR EXP (SETQ EXP (READ]
                                                                    ; So that ADDFILE will have necessary information when it is
                                                                    ; called.
             (FILECREATED1 (CDR EXP))                               ; does error checking on filecreated expression

             )
            ((NULL EXP)
             (SKREAD)))
        (RETURN T)
    YES                                                             ; This IS one of the expressions specified by VARLST.
        [COND
           ((NULL EXP)
```

;; If EXP is non-null, means for some reason it had to be READ, e.g., there was an edit pattern in VARLST. In this case not ;; necessary to SKREAD since we have already passed over that expression.

```
            (SETQ EXP (READ]
        [COND
           ((AND (NEQ VARLST 'LOADFROM)
                 (NEQ VARLST 'LOADCOMP))
            (SETQ DONELST (CONS EXP DONELST]
        (COND
           ((AND (NEQ LDFLG 'EXPRESSIONS)
                 (NEQ LDFLG 'GETDEF))
            (EVAL EXP)))
        (RETURN T])
```

(**SCANDECLARECOLON**

```
[LAMBDA (EVALFLG DONTCOPIES)                                        (* bvm%: "30-Aug-86 16:06")
```

;; handles DECLARE:'s only called for either VARS=COMP, or for looking for specific expression or expresions, e.g. VARS, or edit pattern. For ;; EXPRESSIONS, T, etc., higher call to SCANEXP has already decided what to do.

```
(PROG ((VARLST (if DONTCOPIES
                   then T
                 else VARLST))
       TEM)
  LP   (SETQ ADR (GETFILEPTR))
       [SELECTQ (SETQ TEM (RATOM))
            ((%( %[)
                 (SETQ ADR (SUB1 (GETFILEPTR)))
```

;; reason for this is that there may have been some separators before the (, e.g. a space and c.r., and in this case the ADR will ;; not match up with what was stored in the file map, which would be the position just before the (. The right way to do this is of ;; course not to RATOM but to do a loop with peekc until you ee a non-separator and then record the address. however, thi is ;; inefficient and unnecessary since this is the nly case where it matters

```
                 (SELECTQ (SETQ TEM (RATOM))
                      (DEFINEQ (PROG ((ADR ADR))
                                     (SCANDEFINEQ)
```

;; easier to call scandefineq even if FNS is NIL because it knows how to position file pointer without aving ;; to call skread by using filemap

```
                                      )
                                     [COND
                                        ((AND EVALFLG (EQ VARLST 'LOADCOMP)
                                              (EQ FNLST T))
```

;; LOADCOMP is handled specially. the SCANDEFINEQ would not have actually done any defining, ;; just scanned for the purposes of constructing the map.

```
                                         (SETFILEPTR NIL ADR)
                                         (SETQ TEM (READ))
                                         (COND
                                            ((OR (EQ LDFLG 'EXPRESSIONS)
                                                 (EQ LDFLG 'GETDEF))
                                             (SETQ DONELST (CONS TEM DONELST)))
                                            (T (EVAL TEM])
                      (DECLARE%: (SCANDECLARECOLON EVALFLG DONTCOPIES))
                      (SCANEXP TEM (PROG1 (RATOM)
                                          (SETFILEPTR NIL ADR))
                               EVALFLG)))
            ((%) %])
                 (RETURN T))
            (COND
               (DONTCOPIES (SELECTQ TEM
                                    (DONTCOPY (SETQ EVALFLG T))
                                    ((EVAL@COMPILEWHEN)
                                        (SKREAD))
                                    (COPYWHEN (SKREAD)
                                              (SETQ EVALFLG T))
                                    NIL))
               ((NEQ LDFLG 'GETDEF)                                 ; getdef means ignore tags, find it if its there.
                (SELECTQ TEM
                     ((EVAL@COMPILE DOEVAL@COMPILE)
                         (AND (EQ VARLST 'LOADCOMP)
                              (SETQ EVALFLG T)))
                     (DONTEVAL@COMPILE
                         (AND (EQ VARLST 'LOADCOMP)
```

```
                              (SETQ EVALFLG NIL)))
                         ((EVAL@LOAD DOEVAL@LOAD)
                              (AND (NEQ VARLST 'LOADCOMP)
                                   (SETQ EVALFLG T)))
                         (DONTEVAL@LOAD
                              (AND (NEQ VARLST 'LOADCOMP)
                                   (SETQ EVALFLG NIL)))
                         (EVAL@COMPILEWHEN
                              (SETQ TEM (READ))
                              (AND (EQ VARLST 'LOADCOMP)
                                   (SETQ EVALFLG (EVAL TEM))))
                         (EVAL@LOADWHEN
                              (SETQ TEM (READ))
                              (AND (NEQ VARLST 'LOADCOMP)
                                   (SETQ EVALFLG (EVAL TEM))))
                         (COPYWHEN (SKREAD))
                         NIL]
              (GO LP])
```

(**SCANFILE1**

```
[LAMBDA (DICT LST)                                          ; Edited 16-Apr-2018 17:37 by rmk:
  (AND (NULL LST)
       (SETQ LST FNLST))
```

;; looks up functions on LST, if given, but removes them from FNLST. This so can be called directly from LOADFNS.

```
(PROG ((DICTTAIL DICT)
       X FNAME TEM)
  $$LP
       (COND
          ((OR (NLISTP DICTTAIL)
               (NOT LST))
           (RETURN NIL)))
       (SETQ X (CAR DICTTAIL))
```

;; X = map entry. For compiled definitions is (start end . fn). For source files, it's (start end . triples), where each triple is (fn start . end).

```
       (COND
          [(NLISTP (SETQ FNAME (CDDR X)))                   ; compiled definition.
           (COND
              ((OR (EQ LDFLG 'EXPRESSIONS)
                   (EQ LDFLG 'GETDEF)
                   (EQ VARS 'LOADCOMP))                     ; User wants symbolic definitions only.

               )
              ([OR (EQ LST T)
                   (MEMB FNAME LST)
                   (SOME LST (FUNCTION (LAMBDA (Y)
                                       (TMPSUBFN FNAME Y]   ; User wants all functions, this one in particular, or this is a subfn
                                                            ; of a desired fn
               (SETFILEPTR NIL (CAR X))
               (COND
                  ([NOT (OR (EQ (SETQ TEM (READ))
                                'BINARY)
                            (GETPROP TEM 'CODEREADER]
```

;; a file map was built in core, but it isnt right, e.g. user ftped another file by same name since this map was built in core.
;; so remove map and retry

```
                   (RETRYSCAN)))
               (SETFILEPTR NIL (CAR X))
               (LAPRD FNAME)
               (SCANFILE2 FNAME]
          (T                                                ; DEFINEQ or DEFUN.
             (for Y DEFUNFLG in (CDDR X) do (SETQ DEFUNFLG NIL)
                                            [COND
                                               [(EQ VARS 'LOADCOMP)
                                                (AND (NOT (FMEMB (CAR Y)
                                                                 NOFIXFNSLST))
                                                     (SETQ NOFIXFNSLST (CONS (CAR Y)
                                                                             NOFIXFNSLST]
                                               ((OR (EQ LST T)
                                                    (MEMB (CAR Y)
                                                          LST))
                                                (SETFILEPTR NIL (CADR Y))
                                                (COND
                                                   ([NOT (OR [EQ (CAR Y)
                                                                 (CAR (SETQ TEM (READ]
                                                             (SETQ DEFUNFLG (AND (EQ (CAR TEM)
                                                                                     'CL:DEFUN)
                                                                                 (EQ (CAR Y)
                                                                                     (CADR TEM]
                                                    (ERROR '"filemap does not agree with contents of" (INPUT)
                                                           T)))
                                                (COND
                                                   ((OR (EQ LDFLG 'EXPRESSIONS)
                                                        (EQ LDFLG 'GETDEF))
                                                    (SCANFILE2 TEM))
                                                   (DEFUNFLG (IF (MEMB LDFLG '(PROP ALLPROP))
```

```
                                                                    THEN (PUTDEF (CADR TEM)
                                                                                 'FUNCTIONS TEM)
                                                                ELSE (EVAL TEM))
                                                            (SCANFILE2 (CAR Y)))
                                                    (T (DEFINE (LIST TEM))
                                                       (SCANFILE2 (CAR Y]
                    while LST)))
          (SETQ DICTTAIL (CDR DICTTAIL))
          (GO $$LP])
```

(SCANFILE2

```
  [LAMBDA (X)
    (SETQ DONELST (CONS X DONELST))
    (AND (NEQ FNLST T)
         (SETQ FNLST (DREMOVE (COND
                                 ((LISTP X)
                                  (CAR X))
                                 (T X))
                              FNLST])
```

(TMPSUBFN

```
  [LAMBDA (X FN)                                                    (* bvm%: "28-Aug-86 14:13")
    ;; This guy wants names like FNAnnnnAmmmm...
    (PROG ((N (STRPOS FN X 1 NIL T T))
           NX C)
          (if (OR (NULL N)
                  (NEQ (IREMAINDER (IDIFFERENCE (SETQ NX (ADD1 (NCHARS X)))
                                                N)
                                   5)
                       0))
              then                                                  ; X does not start with FN, or end in an integral number of 5
                                                                    ; character pieces
                  (RETURN))
      LP  (if [OR (NEQ (NTHCHARCODE X N)
                       (CHARCODE A))
                  (NOT (for I from 1 to 4 always (AND (SETQ C (NTHCHARCODE X (IPLUS I N)))
                                                      (IGEQ C (CHARCODE 0))
                                                      (ILEQ C (CHARCODE 9]
              then (RETURN)
            elseif (IGEQ (add N 5)
                         NX)
              then (RETURN T))
          (GO LP])
```

(RETRYSCAN

```
  [LAMBDA NIL                                                       (* bvm%: "28-Aug-86 17:05")
    (COND
       ((GETHASH FILE *FILEMAP-HASH*)
        (REMHASH FILE *FILEMAP-HASH*)
        (PRIN1 "something is wrong with the filemap for " T)
        (PRINT FILE T)
        (PRIN1 "rebuilding map..." T)
        (RETFROM 'LOADFNSCAN (LOADFNSCAN)))
       (T (SCANFILEHELP])
```

(SCANFILEHELP

```
  [LAMBDA NIL
    ;; Edited  2-May-2022 11:37 by larry: used to suggest contacting 1100 support (medley issue #411)
    ;;  this used to suggest contacting 1100 support (medley issue #411)   ; Edited  2-May-2022 11:31 by larry
                                                                           (* JonL "15-Dec-83 21:04")
    ;; This function used to spit out a 'sermon' about sysouting and informing W.  Teitelman.
    (PRIN1 '"something is wrong with either the filemap or format of " T)
    (PRIN1 (INPUT)
           T)
    (PRINTOUT T '"Here are some possibilities:" T "(1) you edited the file with a text editor;" T "(2) you
           printed a DEFINEQ in the file directly, i.e. without using the FNS command;" T "(3) the file got
           clobbered." T)
    (PRIN1
           '"Note: for (1) and (2), you may still be able to use this file by setting USEMAPFLG to NIL and then
            reexecuting the operation that caused this message." T)
    (TERPRI)
    (HELP])

)

(RPAQQ NOT-FOUNDTAG NOT-FOUND%:)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS LASTWORD LOADOPTIONS SYSFILES NOT-FOUNDTAG)
```

```
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY

(BLOCK%: SCANFILEBLOCK (ENTRIES LOADFNSCAN TMPSUBFN SCANFILE1)
       LOADFNSCAN SCANFILE0 SCANCOMPILEDFN SCANDEFINEQ SCANEXP SCANDECLARECOLON SCANFILE1 SCANFILE2 TMPSUBFN
       (LOCALFREEVARS FNADRLST DICT DICT0 ADR)
       (SPECVARS VARLST)
       (RETFNS SCANFILE0))
)

(PUTPROPS LOADFNS COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1986 1987 1989 1990 2018 2022))
```

FUNCTION INDEX

VARIABLE INDEX