File created: 1-May-2024 14:53:20 **{WMEDLEY}<lispusers>COMPAREDIRECTORIES.;260**

edit by: rmk

changes to: (FNS COMPAREDIRECTORIES)

previous date: 26-Mar-2024 21:42:47 {WMEDLEY}<lispusers>COMPAREDIRECTORIES.;259

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ COMPAREDIRECTORIESCOMS
       [;; Compare the contents of two directories.
        (FNS COMPAREDIRECTORIES COMPAREDIRECTORIES.INFOS COMPAREDIRECTORIES.CANDIDATES CDENTRIES.SELECT
             COMPAREDIRECTORIES.INFOS.TYPE MATCHNAME CD.INSURECDVALUE CD.UPDATEWIDTHS)
        (FNS CDFILES CDFILES.MATCH CDFILES.PATS)
        (FNS CDPRINT CDPRINT.HEADER CDPRINT.LINE CDPRINT.MAXWIDTHS CDPRINT.COLHEADERS CDPRINT.COLUMNS CDTEDIT)
        (FNS CDMAP CDENTRY CDSUBSET CDMERGE CDMERGE.COMMON CD.SORT)
        (FNS BINCOMP EOLTYPE EOLTYPE.SHOW)
        (RECORDS CDMAXNCHARS CDVALUE CDENTRY CDINFO)
        ;; look for compiled files older than the sources
        (FNS FIND-UNCOMPILED-FILES FIND-UNSOURCED-FILES FIND-SOURCE-FILES FIND-COMPILED-FILES
             FIND-UNLOADED-FILES FIND-LOADED-FILES FIND-MULTICOMPILED-FILES)
        (FNS CREATED-AS SOURCE-FOR-COMPILED-P COMPILE-SOURCE-DATE-DIFF)
        (FNS FIX-DIRECTORY-DATES FIX-EQUIV-DATES COPY-COMPARED-FILES COPY-MISSING-FILES COMPILED-ON-SAME-SOURCE)
        [VARS (ONESECOND (IDIFFERENCE (IDATE "1-Jan-2020 12:00:01")
                                      (IDATE "1-Jan-2020 12:00:00"]
        (INITVARS (LASTCDVALUE NIL))
        ;; Compare-directories browser
        (COMS (FNS CDBROWSER CDBROWSER.STRINGS)
              ;; TABLEBROWSER browser
              (FILES (SYSLOAD)
                     TABLEBROWSER)
              (DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP)
                                                      TABLEBROWSER))
              (FNS CD.TABLEITEM CD.TABLEITEM.PRINTFN CD.TABLEITEM.COPYFN CDTABLEBROWSER.HEADING.REPAINTFN)
              (FNS CDTABLEBROWSER.WHENSELECTEDFN CD.COMMANDSELECTEDFN CD-MENUFN CD-COMPARE-FILES CDBROWSER-COPY
                   CDBROWSER-DELETE-FILE CD-SWAPDIRS)
              (VARS CDTABLEBROWSER.MENUITEMS)
              (FILES (SYSLOAD)
                     COMPARESOURCES COMPARETEXT)
              (P (MOVD? 'NILL 'TEDIT.FILEDATE])
```

;; Compare the contents of two directories.

```
(DEFINEQ
```

(COMPAREDIRECTORIES

```
  [LAMBDA (DIR1 DIR2 SELECT INCLUDEDFILES EXCLUDEDFILES USEDIRECTORYDATE OUTPUTFILE ALLVERSIONS
              FIXDIRECTORYDATES SHORTDIRNAMES)                    ; Edited  1-May-2024 14:52 by rmk
                                                                  ; Edited 29-Sep-2023 17:25 by rmk
                                                                  ; Edited  5-Apr-2023 10:12 by rmk
                                                                  ; Edited 29-Mar-2022 11:50 by rmk
                                                                  ; Edited 23-Feb-2022 21:10 by rmk
                                                                  ; Edited  4-Jan-2022 12:09 by rmk
                                                                  ; Edited 31-Oct-2021 11:01 by rmk:
                                                                  ; Edited  7-Jan-2021 23:21 by rmk:
    ;; Compare the contents of two directories, e.g., for change-control purposes.  Compares files matching FILEPATTERN (or *.*;) on DIR1 and DIR2,
    ;; listing which is newer, or when one is not found on the other.  If SELECT is or contains SAME/=, BEFORE/<, AFTER/>, then files where DIR1 is
    ;; the same as, earlier than, or later than DIR2 are selected. SELECT= NIL is the same as (< >), T is the same as (< > =).  Also allows selection
    ;; based on file-length criteria.
    ;;
    ;; Unless USEDIRECTORYDATE, comparison is with respect to the the LISP filecreated dates if available.
    ;;
    ;; If OUTPUTFILE is NIL, the list of compared entries is returned.  Otherwise the selected entries are printed on OUTPUTFILE (T for the display).
    [SETQ SELECT (SELECTQ SELECT
                      (NIL '(< > -* *-))
                      (T '(< > -* *- =))
                      (for S in (MKLIST SELECT) collect (SELECTQ S
                                                                ((AFTER >)
                                                                    '>)
                                                                ((BEFORE <)
                                                                    '<)
                                                                ((SAME SAMEDATE =)
                                                                    '=)
                                                                (AUTHOR 'AUTHOR)
                                                                (-* '-*)
```

```
                                                    (*- '*-)
                                                    (~= '~=)
                                                    (ERROR "UNRECOGNIZED SELECT PARAMETER" S]
    (PROG (INFOS1 INFOS2 CDENTRIES DEPTH1 DEPTH2 CDVALUE (DATE (DATE)))
```

;; DIRECTORYNAME here to get unrelativized specifications for header.

;; Allow all subdirectories if a directory ends in *, but peel it off for the resolution

```
      (CL:WHEN (EQ '* (NTHCHAR DIR1 -1))
          (SETQ DEPTH1 T)
          (SETQ DIR1 (SUBSTRING DIR1 1 -2)))
      (CL:WHEN (EQ '* (NTHCHAR DIR2 -1))
          (SETQ DEPTH2 T)
          (SETQ DIR2 (SUBSTRING DIR2 1 -2)))
      (SETQ DIR1 (OR (DIRECTORYNAME (OR DIR1 T))
                     DIR1))
      (SETQ DIR2 (OR (DIRECTORYNAME (OR DIR2 T))
                     DIR2))
      (CL:WHEN FIXDIRECTORYDATES
          (PRINTOUT T "Fixing directory dates" T)
          (FIX-DIRECTORY-DATES DIR1)
          (FIX-DIRECTORY-DATES DIR2))
      (CDPRINT.HEADER (OR (CAR SHORTDIRNAMES)
                          DIR1)
             (OR (CADR SHORTDIRNAMES)
                 DIR2)
             SELECT DATE T)
      (PRINTOUT T " ... ")
      (SETQ INFOS1 (COMPAREDIRECTORIES.INFOS DIR1 INCLUDEDFILES EXCLUDEDFILES ALLVERSIONS DEPTH1
                          USEDIRECTORYDATE (MEMB 'AUTHOR SELECT)))
      (SETQ INFOS2 (COMPAREDIRECTORIES.INFOS DIR2 INCLUDEDFILES EXCLUDEDFILES ALLVERSIONS DEPTH2
                          USEDIRECTORYDATE))
```

;; The CAR of each info is the atomic match-name, the CDR is a list of infos with that matchname, only 1 unless AllVERSIONS.

```
      (SETQ CDVALUE (CREATE CDVALUE
                            CDDIR1 _ DIR1
                            CDDIR2 _ DIR2
                            CDCOMPAREDATE _ DATE
                            CDSELECT _ SELECT))
      (CL:UNLESS (OR INFOS2 INFOS1)
             (RETURN CDVALUE))
```

;; Correlate the I1's and I2's with the same match name, then do the select filtering and insert the date relations

```
      (SETQ CDENTRIES (SORT (CDENTRIES.SELECT (COMPAREDIRECTORIES.CANDIDATES INFOS1 INFOS2)
                                              SELECT)
                            (FUNCTION CD.SORT)))
      (PRINTOUT T (LENGTH CDENTRIES)
             " entries" T)
      (REPLACE CDENTRIES OF CDVALUE WITH CDENTRIES)
      (CD.UPDATEWIDTHS CDVALUE)
      (SETQ LASTCDVALUE CDVALUE)
      (CL:UNLESS OUTPUTFILE (RETURN CDVALUE))
      (RETURN (CDPRINT CDVALUE OUTPUTFILE NIL (MEMB 'AUTHOR SELECT])
```

(COMPAREDIRECTORIES.INFOS

```
  [LAMBDA (DIR INCLUDEDFILES EXCLUDEDFILES ALLVERSIONS DEPTH USEDIRECTORYDATE INCLUDEAUTHOR)
```

;; Edited 29-Sep-2023 17:25 by rmk

;; Edited 22-May-2022 14:17 by rmk

;; Edited 29-Mar-2022 11:53 by rmk: Produces a list of CDINFOS with the match-name consed on to the front.

;; Each entry is a list of the form (matchname . CDINFOS). CDINFOS is guaranteed to be a singleton, unless ALLVERSIONS.

```
    (FOR FULLNAME TYPE LDATE STREAM (STARTPOS _ (ADD1 (NCHARS DIR)))
       IN (CDFILES DIR INCLUDEDFILES EXCLUDEDFILES ALLVERSIONS DEPTH)
       COLLECT
```

;; GDATE/IDATE in case Y2K

```
            (SETQ STREAM (OPENSTREAM FULLNAME 'INPUT))          ; So all the GETFILEINFO's, FILEDATES, etc. don't have to do
                                                                ; the directory searching.
                                                                ; Is it a Lisp file? Get it's internal filecreated date.
            (CL:MULTIPLE-VALUE-SETQ (TYPE LDATE)
                   (COMPAREDIRECTORIES.INFOS.TYPE STREAM))
            (PROG1 (LIST (MATCHNAME FULLNAME STARTPOS)
                         (CREATE CDINFO
                                 FULLNAME _ (FULLNAME STREAM)
                                 DATE _ (GDATE (IDATE (IF (OR USEDIRECTORYDATE (NULL LDATE))
                                                          THEN (GETFILEINFO STREAM 'CREATIONDATE)
                                                        ELSE (SETFILEINFO STREAM 'CREATIONDATE LDATE)
                                                             LDATE)))
                                 LENGTH _ (GETFILEINFO STREAM 'LENGTH)
                                 AUTHOR _ (AND INCLUDEAUTHOR (GETFILEINFO STREAM 'AUTHOR))
                                 TYPE _ TYPE
                                 EOL _ (EOLTYPE STREAM)))
                   (CLOSEF? STREAM))
       FINALLY
```

;; Sort to get all entries with the same matchname adjacent. Presumably we would only need to collect multiples if ALLVERSIONS, but

```
;; in a case-sensitive file system we might see files with names that differ in case.  We have deliberately given them a case-insensitive
;; matchname, so we  can expose this issue in the display.

;; If we see (MN X)(MN Y), smash the Y in after the X

(RETURN (FOR ITAIL I VAL MN ON (SORT $$VAL T) DO (SETQ I (CAR ITAIL))
                                                 (SETQ MN (CAR I))
                                                 [WHILE (EQ MN (CAADR ITAIL))
                                                    DO (POP ITAIL)
                                                       (PUSH (CDR I)
                                                             (CADR (CAR ITAIL]
                                                 (PUSH VAL I)
                FINALLY (RETURN (DREVERSE VAL])
```

(COMPAREDIRECTORIES.CANDIDATES

```
[LAMBDA (INFOS1 INFOS2)

    ;; Edited 24-Feb-2022 10:00 by rmk: Correlate the I1's and I2's with the same match name. Rely on the fact that the lists are sorted.

    (SETQ INFOS1 (SORT INFOS1 T))
    (SETQ INFOS2 (SORT INFOS2 T))
    (LET (PAIRS)
         (BIND I1 I2 (I1TAIL _ INFOS1)
               (I2TAIL _ INFOS2) DO (IF (AND I1TAIL I2TAIL)
                                        THEN (SETQ I1 (CAR I1TAIL))
                                             (SETQ I2 (CAR I2TAIL))
                                             (IF (EQ (CAR I1)
                                                     (CAR I2))
                                                 THEN (PUSH PAIRS (LIST (CAR I1)
                                                                        (CDR I1)
                                                                        (CDR I2)))
                                                      (POP I1TAIL)
                                                      (POP I2TAIL)
                                               ELSEIF (ALPHORDER (CAR I1)
                                                                 (CAR I2))
                                                 THEN (PUSH PAIRS (LIST (CAR I1)
                                                                        (CDR I1)
                                                                        (CONS NIL)))
                                                      (POP I1TAIL)
                                                 ELSE (PUSH PAIRS (LIST (CAR I2)
                                                                        (CONS NIL)
                                                                        (CDR I2)))
                                                      (POP I2TAIL))
                                      ELSEIF I1TAIL
                                        THEN [FOR I1 IN I1TAIL DO (PUSH PAIRS (LIST (CAR I1)
                                                                                    (CDR I1)
                                                                                    (CONS NIL]
                                             (RETURN)
                                      ELSEIF I2TAIL
                                        THEN [FOR I2 IN I2TAIL DO (PUSH PAIRS (LIST (CAR I2)
                                                                                    (CONS NIL)
                                                                                    (CDR I2]
                                             (RETURN)
                                      ELSE (RETURN)))
        ;; Take the cross products (if ALLVERSIONS) to create a list of (MN I1 I2) CDENTRIES with singleton infos.
        (FOR P MN CANDIDATES IN PAIRS
           DO (SETQ MN (CAR P))
              [FOR I1 IN (CADR P)
                 DO (FOR I2 IN (CADDR P)
                       DO (PUSH CANDIDATES (CREATE CDENTRY
                                                   MATCHNAME _ MN
                                                   INFO1 _ I1
                                                   INFO2 _ I2]
           FINALLY (RETURN CANDIDATES])
```

(CDENTRIES.SELECT

```
[LAMBDA (CANDIDATES SELECT)                                  ; Edited 23-Feb-2022 20:45 by rmk
                                                             ; Edited  4-Jan-2022 21:31 by rmk

    ;; Does the pairwise select filter and inserts the date relation

    (for CDE MATCHNAME INFO1 INFO2 IDATE1 IDATE2 DATEREL BINCOMP [COMPAREDATE _ (INTERSECTION SELECT
                                                                                  '(< > =]
       in CANDIDATES eachtime (SETQ MATCHNAME (FETCH (CDENTRY MATCHNAME) OF CDE))
                              (SETQ INFO1 (FETCH (CDENTRY INFO1) OF CDE))
                              (SETQ INFO2 (FETCH (CDENTRY INFO2) OF CDE))
                              (if (AND INFO1 INFO2)
                                  then (SETQ IDATE1 (IDATE (fetch DATE of INFO1)))
                                       (SETQ IDATE2 (IDATE (fetch DATE of INFO2)))
                                       (SETQ DATEREL (if (IGREATERP IDATE1 IDATE2)
                                                         then '>
                                                       elseif (ILESSP IDATE1 IDATE2)
                                                         then '<
                                                         else '=))
                                else ;; Just for printing--no comparison
                                     (SETQ DATEREL '*))
```

```
    when (if (AND INFO1 INFO2)
             then (CL:WHEN (OR (NULL COMPAREDATE)
                               (SELECTQ DATEREL
                                    (> (MEMB '> COMPAREDATE))
                                    (< (MEMB '< COMPAREDATE))
                                    (= (MEMB '= COMPAREDATE))
                                    (SHOULDNT)))
                           (SETQ BINCOMP (BINCOMP (fetch (CDINFO FULLNAME) OF INFO1)
                                                  (fetch (CDINFO FULLNAME) OF INFO2)
                                                  T
                                                  (fetch (CDINFO EOL) OF INFO1)
                                                  (fetch (CDINFO EOL) OF INFO2)))
                           (CL:WHEN (EQ T BINCOMP)
                               ;; Byte-equivalent files with different dates.  Presumably the earlier date is more accurate, move back the date of
                               ;; the later file and make DATEREL be =.  Perhaps we should do this even if there is only an EOL difference
                               ;; (BINCOMP non-NIL).;; Byte-equivalent files with different dates.  Presumably the earlier date is more accurate,
                               ;; move back the date of the earlier file and make DATEREL be =.  Perhaps we should do this even if there is only
                               ;; an EOL difference (BINCOMP non-NIL).

                               ;; We do this even if FIXDIRECTORYDATES is false, that addresses a property of individual Lisp source files.

                               (SELECTQ DATEREL
                                    (> (SETFILEINFO (FETCH (CDINFO FULLNAME) OF INFO1)
                                                    'CREATIONDATE
                                                    (REPLACE (CDINFO DATE) OF INFO1 WITH (FETCH (CDINFO DATE)
                                                                                                OF INFO2))))
                                    (< (SETFILEINFO (FETCH (CDINFO FULLNAME) OF INFO2)
                                                    'CREATIONDATE
                                                    (REPLACE (CDINFO DATE) OF INFO2 WITH (FETCH (CDINFO DATE)
                                                                                                OF INFO1))))
                                    NIL)
                               (SETQ DATEREL '=))
                           ;; We want the ~= test to reflect exact byte equivalence, including the same EOL.  We use the BINCOMP value below
                           ;; to indicate EOL differences, so we check it here.
                           [NOT (AND (MEMB '~= SELECT)
                                     BINCOMP
                                     (EQ (fetch (CDINFO EOL) OF INFO1)
                                         (fetch (CDINFO EOL) OF INFO2])
         elseif INFO1
           then ;; OK if INFO2 is missing?
                (MEMB '*- SELECT)
           else ;; OK if INFO1 is missing?
                (MEMB '-* SELECT))
    collect (REPLACE (CDENTRY DATEREL) OF CDE WITH DATEREL)
            (REPLACE (CDENTRY EQUIV) OF CDE WITH (CL:UNLESS (EQ DATEREL '*)
                                                            BINCOMP))
            CDE])
```

(COMPAREDIRECTORIES.INFOS.TYPE

```
[LAMBDA (FILE)                                                  ; Edited 28-Sep-2023 23:09 by rmk
                                                                ; Edited 22-May-2022 14:27 by rmk
                                                                ; Edited 25-Apr-2022 09:02 by rmk
                                                                ; Edited  4-Jan-2022 13:10 by rmk
                                                                ; Edited 12-Dec-2021 22:50 by rmk
  (LET (TYPE DATE)
       (CL:MULTIPLE-VALUE-SETQ (TYPE DATE)
              (LISPFILETYPE FILE))
       (CL:UNLESS TYPE
           (SETQ TYPE (IF (SETQ DATE (TEDIT.FILEDATE FILE))
                          THEN 'TEDIT
                        ELSEIF (PRINTFILETYPE FILE)
                        ELSE (MEMB (FILENAMEFIELD FILE 'EXTENSION)
                                   '(TXT TEXT SH MD C))
                          THEN 'TEXT
                        ELSE 'OTHER)))
       (CL:VALUES TYPE DATE])
```

(MATCHNAME

```
[LAMBDA (NAME STARTPOS)                                         ; Edited 24-Feb-2022 09:10 by rmk
                                                                ; Edited 23-Dec-2021 22:41 by rmk
                                                                ; Edited  5-Sep-2020 13:41 by rmk:
  ;; The canonical name for matching related files
  (LET [(M (UNSLASHIT (U-CASE (PACKFILENAME 'VERSION NIL 'BODY (SUBATOM NAME STARTPOS]
       ;; Strip off the nuisance period
       (CL:IF (EQ (CHARCODE %.)
                  (NTHCHARCODE M -1))
           (SUBATOM M 1 -2)
           (MKATOM M))])
```

(CD.INSURECDVALUE

```
  [LAMBDA (CDVALUE?)                                                    ; Edited 30-Nov-2021 14:37 by rmk:
    ;; Maybe just a list of entries without the global information.  Try to fix it
    (CL:UNLESS CDVALUE?
        (PRINTOUT T T "Note:  Using LASTCDVALUE" T T)
        (SETQ CDVALUE? LASTCDVALUE))
    (CD.UPDATEWIDTHS (IF (STRINGP (FETCH (CDVALUE CDDIR2) OF CDVALUE?))
                         THEN CDVALUE?
                       ELSE (create CDVALUE
                                    CDENTRIES _ CDVALUE?
                                    CDDIR1 _ [for E in CDVALUE? when (fetch INFO1 of E)
                                                do (RETURN (PACKFILENAME 'NAME NIL 'EXTENSION NIL 'VERSION NIL
                                                                         'BODY
                                                                         (fetch (CDINFO FULLNAME)
                                                                            OF (fetch INFO1 of E]
                                    CDDIR2 _ [for E in CDVALUE? when (fetch INFO2 of E)
                                                do (RETURN (PACKFILENAME 'NAME NIL 'EXTENSION NIL 'VERSION NIL
                                                                         'BODY
                                                                         (fetch (CDINFO FULLNAME)
                                                                            OF (fetch INFO2 of E]
                                    CDCOMPAREDATE _ (DATE])
```

(CD.UPDATEWIDTHS

```
  [LAMBDA (CDVALUE)                                                     ; Edited  4-Dec-2021 09:25 by rmk
                                                                        ; Edited 30-Nov-2021 13:34 by rmk:
    (LET ((WIDTHS (CDPRINT.MAXWIDTHS CDVALUE)))
         (REPLACE (CDVALUE CDMAXNC1) OF CDVALUE WITH (CAR WIDTHS))
         (REPLACE (CDVALUE CDMAXNC2) OF CDVALUE WITH (CADR WIDTHS)))
    CDVALUE])

)

(DEFINEQ
```

(CDFILES

```
  [LAMBDA (DIR INCLUDEDFILES EXCLUDEDFILES ALLVERSIONS DEPTH)           ; Edited 17-Jun-2023 23:04 by rmk
                                                                        ; Edited  3-Oct-2022 12:03 by rmk
                                                                        ; Edited 25-Apr-2022 08:42 by rmk
                                                                        ; Edited  5-Mar-2022 15:05 by rmk
                                                                        ; Edited 16-Oct-2020 13:42 by rmk:
    ;; Returns a list of fullnames for files that satisfy the criteria.  We generate all candidates that match INCLUDEDFILES but not EXCLUDEDFILES in
    ;; DIR.

    ;; For each name returned by (DIRECTORY DIR), assumes that INCLUDEDFILES applies to the suffix after the directory (i.e. after NAMEPOS).
    ;; That includes possibly subdirectories, dotted files in ultimate file names, and versions.

    ;;    Exclude subdirectories unless INCLUDEDFILES includes *>*

    ;;    Exclude dotted files (.xxx) unless INCLUDEDFILES includes .*

    ;;    Exclude older versions unless ALLVERSIONS=T

    ;;  DEPTH is the number of subdirectories below the ones specified in DIR (NIL top-level of DIR only, T = any depth)

    ;; Resolve relative directories, so we can suppress subdirectory matches.

    ;; EXCLUDEDFILES is a filepattern with * meaning everything, COM means *.LCOM and *.DFASL

    [SETQ EXCLUDEDFILES `(*>.DS_Store
                          ,@(MKLIST EXCLUDEDFILES]
    (CL:UNLESS (EQMEMB '.* INCLUDEDFILES)                               ; Excluded dot files unless specifically asked for
        [SETQ EXCLUDEDFILES `(.* ,@(MKLIST EXCLUDEDFILES])
    (SETQ EXCLUDEDFILES (LDIFFERENCE EXCLUDEDFILES INCLUDEDFILES))
    (LET ([INCLUDES (CDFILES.PATS (OR INCLUDEDFILES '*.*]
          (EXCLUDES (AND EXCLUDEDFILES (CDFILES.PATS EXCLUDEDFILES)))
          (*UPPER-CASE-FILE-NAMES* NIL)
          HOST ENUMPAT)
         (SETQ HOST (FILENAMEFIELD.STRING DIR 'HOST))
         (SETQ DIR (FILENAMEFIELD.STRING DIR 'DIRECTORY))
         (CL:UNLESS DEPTH
             ;; DEPTH is the number of internal > or /
             (SETQ DEPTH (FOR P IN INCLUDES LARGEST (CADDDR P) FINALLY (RETURN $$EXTREME))))
         ;; ENUMPAT is the single pattern that we use for the directory enumeration (given the enumeration depth). We have to go to the most
         ;; general specification, then filter the generated results.
         (FOR P (N _ (CAAR INCLUDES))
               (E _ (CADAR INCLUDES))
               (SD _ (CADDAR INCLUDES)) IN (CDR INCLUDES) DO (CL:UNLESS (EQ '* N)
                                                                 (SETQ N (POP P)))
                                                             (CL:UNLESS (EQ '* E)
                                                                 (SETQ E (POP P)))
                                                             (CL:UNLESS (OR (EQ SD '*)
                                                                            (EQ SD (CAR P)))
                                                                    (SETQ SD NIL))
            FINALLY (CL:WHEN (EQ SD '*)
                             (SETQ SD ""))
                 ;; If We don't prefix TOPDIR with <, then if TOPDIR contains a colon it is interpreted as a device.
                 (SETQ ENUMPAT (PACKFILENAME.STRING 'HOST HOST 'DIRECTORY (CONCAT "<" DIR ">" (OR SD ""))
```

```
                                    'NAME N 'EXTENSION E 'VERSION (CL:IF ALLVERSIONS
                                                                         '*
                                                                         ""))))
                (CL:UNLESS (CDR INCLUDES)                    ; No further filtering if there is only one pattern
                    (SETQ INCLUDES NIL)))
        ;; We enumerate all the files, checking to see that
        (FOR FULLNAME NAME EXT SUBDIR UNPACK THISDEPTH (STARTPOS _ (IPLUS 2 (NCHARS DIR)))
           IN (DIRECTORY ENUMPAT `(DEPTH ,DEPTH COLLECT)
                     NIL
                     (CL:IF ALLVERSIONS
                         "*"
                         ""))
           EACHTIME (SETQ UNPACK (UNPACKFILENAME FULLNAME))
                 (SETQ NAME (LISTGET UNPACK 'NAME))
                 (SETQ EXT (LISTGET UNPACK 'EXTENSION))
                 (CL:UNLESS NAME
                     (CL:WHEN EXT                            ; .XY
                         (SETQ NAME (PACK* "." EXT))
                         (SETQ EXT NIL)))
                 (CL:UNLESS (OR NAME EXT)                    ; Must have been a directory
                     (GO $$ITERATE))
                 (SETQ SUBDIR (SUBATOM (LISTGET UNPACK 'DIRECTORY)
                                       STARTPOS))
                 (SETQ THISDEPTH (FOR I (CNT _ 1) FROM 1 DO (SELCHARQ (NTHCHARCODE SUBDIR I)
                                                                      ((> /)
                                                                       (ADD CNT 1))
                                                                      (NIL (RETURN CNT))
                                                                      NIL)))
           WHEN (OR (NULL INCLUDES)
                    (CDFILES.MATCH SUBDIR NAME EXT THISDEPTH INCLUDES))
           UNLESS (CDFILES.MATCH SUBDIR NAME EXT THISDEPTH EXCLUDES) COLLECT FULLNAME])
```

(CDFILES.MATCH

```
  [LAMBDA (SUBDIR NAME EXT THISDEPTH PATTERNS)               ; Edited 26-Jan-2022 15:33 by rmk
                                                             ; Edited 23-Dec-2021 21:47 by rmk
    ;; True if the components of the fullname match at least one of the patterns
    (THEREIS P IN PATTERNS SUCHTHAT (AND [OR (STRING.EQUAL NAME (CAR P)
                                             FILEDIRCASEARRAY)
                                         (EQ '* (CAR P))
                                         (AND (EQ (CHARCODE %.)
                                                  (CHCON1 (CAR P)))
                                              (EQ (CHARCODE %.)
                                                  (CHCON1 NAME))
                                              (OR (STRING.EQUAL NAME (SUBATOM (CAR P)
                                                                              2))
                                                  (EQ (CHARCODE *)
                                                      (NTHCHARCODE (CAR P)
                                                             2]
                                         (OR (STRING.EQUAL EXT (CADR P))
                                             (EQ '* (CADR P)))
                                         (OR (STRING.EQUAL SUBDIR (CADDR P))
                                             (NULL (CADDR P))
                                             (EQ '* (CADDR P)))
                                         (ILEQ THISDEPTH (CADDDR P])
```

(CDFILES.PATS

```
  [LAMBDA (PATTERNS)                                         ; Edited 17-Jun-2023 23:36 by rmk
                                                             ; Edited 23-Dec-2021 17:02 by rmk
    ;; Returns (NAME EXT SUBDIR DEPTH) items where NAME or EXT may be the wildcard *, SD is the subdirectory (if any) and DEPTH is the
    ;; number of / or > in the subdirectory
    (IF (OR (NULL PATTERNS)
            (EQMEMB '* PATTERNS))
        THEN '(

          (* * NIL 1)
)
      ELSE (FOR P N E SD DEPTH UNPACK INSIDE PATTERNS
              JOIN (SETQ UNPACK (UNPACKFILENAME.STRING P))   ; String so we can tell the difference between x and x.
                   [SETQ SD (MKATOM (LISTGET UNPACK 'SUBDIRECTORY]
                   ;; Count the subdirectory depth
                   [SETQ DEPTH (IF (EQ SD '*)
                                   THEN MAX.SMALLP
                                 ELSE (FOR I (CNT _ 1) FROM 1 DO (SELCHARQ (NTHCHARCODE SD I)
                                                                           ((/ >)
                                                                            (ADD CNT 1))
                                                                           (NIL (RETURN CNT))
                                                                           NIL]
                   (SETQ N (LISTGET UNPACK 'NAME))
                   (SETQ N (if (NULL N)
                               then '*
                             elseif (NEQ 0 (NCHARS N))
```

```
                        then (MKATOM N)))
                  (SETQ E (LISTGET UNPACK 'EXTENSION))
                  (SETQ E (if (NULL E)
                              then '*
                             elseif (NEQ 0 (NCHARS E))
                              then (MKATOM E)))
                  (if [OR (AND (STRING.EQUAL N 'COM)
                               (NULL E))
                          (AND (STRING.EQUAL E 'COM)
                               (MEMB N '                          (* NIL)]
                     THEN (FOR CE IN *COMPILED-EXTENSIONS* COLLECT (LIST '* CE SD DEPTH))
                     ELSE (CONS (IF N
                                    THEN (LIST N E SD DEPTH)
                                  ELSEIF E
                                    THEN
                                        ;; This is the case .XXX, which presumably identifies a dotted file.  If this is supposed to be all files
                                        ;; with extension XXX, it shoud be specified as *.XXX, the case above.  So we move .E into the N
                                        ;; field.
                                         (LIST (PACK* '%. E)
                                               NIL SD DEPTH)
                                  ELSE `

          (* * (\, SD) (\, DEPTH))
])

)

(DEFINEQ
```

(CDPRINT

```
  [LAMBDA (CDVALUE FILE COLHEADINGS PRINTAUTHOR)                    ; Edited 15-Jul-2022 12:03 by rmk
                                                                    ; Edited 26-Jan-2022 13:43 by rmk
                                                                    ; Edited 19-Dec-2021 20:10 by rmk
                                                                    ; Edited 30-Nov-2021 20:59 by rmk:
                                                                    ; Edited 13-Oct-2020 08:38 by rmk:
    ;; Typically CDVALUE will have a provdenance header.  If not, we fake one up, at least for the directories and today's date.
    (SETQ CDVALUE (CD.INSURECDVALUE CDVALUE))
    (RESETLST
        (LET* [STREAM (COLUMNS (CDPRINT.COLUMNS CDVALUE COLHEADINGS PRINTAUTHOR))
                      (DATE1POS (POP COLUMNS))
                      (ENDDATE1 (POP COLUMNS))
                      (COL1WIDTH (POP COLUMNS))
                      (COL2WIDTH (POP COLUMNS))
                      (COL2START (POP COLUMNS))
                      (LENGTH2END (POP COLUMNS))
                      (NCHARSDIR1 (FETCH NCDIR OF (FETCH CDMAXNC1 OF CDVALUE)))
                      (NCHARSDIR2 (FETCH NCDIR OF (FETCH CDMAXNC2 OF CDVALUE]
              (CL:UNLESS (SETQ STREAM (GETSTREAM FILE 'OUTPUT T))
                  [RESETSAVE (SETQ STREAM (OPENSTREAM (PACKFILENAME 'EXTENSION 'TXT 'BODY FILE)
                                                      'OUTPUT
                                                      'NEW))
                             '(PROGN (CLOSEF? OLDVALUE])
              (LINELENGTH 1000 STREAM)                              ; Don't wrap
              (CL:WHEN (FETCH (CDVALUE CDDIR1) OF CDVALUE)
                  (CDPRINT.HEADER CDVALUE STREAM)
                  (PRINTOUT STREAM -2 (LENGTH (fetch CDENTRIES of CDVALUE))
                         " entries" T T))
              (if (fetch CDENTRIES of CDVALUE)
                  then (CDPRINT.COLHEADERS STREAM COLHEADINGS ENDDATE1 COL1WIDTH COL2START COL2WIDTH)
                       (for E in (fetch CDENTRIES of CDVALUE)
                          do (CDPRINT.LINE STREAM E PRINTAUTHOR DATE1POS ENDDATE1 NCHARSDIR1 NCHARSDIR2
                                           LENGTH2END))
                else (PRINTOUT T "CDVALUE is empty" T))
              (AND STREAM (CLOSEF? STREAM))))])
```

(CDPRINT.HEADER

```
  [LAMBDA (DIR1 DIR2 SELECT DATE STREAM)                            ; Edited 26-Jan-2022 13:36 by rmk
    (CL:WHEN (LISTP DIR1)
        ;; A CDVALUE
        (CL:UNLESS STREAM (SETQ STREAM DIR2))
        (SETQ DIR2 (FETCH CDDIR2 OF DIR1))
        (SETQ SELECT (FETCH CDSELECT OF DIR1))
        (SETQ DATE (FETCH CDCOMPAREDATE OF DIR1))
        (SETQ DIR1 (FETCH CDDIR1 OF DIR1)))
    (CL:WHEN DIR1
        (PRINTOUT STREAM "Comparing ")
        (PRINTOUT STREAM DIR1 %# (CL:WHEN (IGREATERP (IPLUS (NCHARS DIR1)
                                                            (NCHARS DIR2))
                                                     70)
                                          (TAB 5))
               " vs. " DIR2)
        (PRINTOUT STREAM T 3 "as of " DATE)
        (CL:WHEN SELECT (PRINTOUT STREAM " selecting " SELECT)))])
```

(CDPRINT.LINE

```
[LAMBDA (STREAM ENTRY PRINTAUTHOR DATE1POS ENDDATE1 NCHARSDIR1 NCHARSDIR2 LENGTH2END)
                                                                        ; Edited 16-Jul-2022 10:19 by rmk
                                                                        ; Edited 22-Nov-2021 22:38 by rmk:
                                                                        ; Edited  9-Jan-2021 10:12 by rmk:
    ;; Format one line of the directory comparison listing.  If PRINTAUTHOR and AUTHOR1 or AUTHOR2 are non-NIL, list the author in parens;
    ;; otherwise omit it.
    (LET ((INFO1 (fetch INFO1 of ENTRY))
          (INFO2 (fetch INFO2 of ENTRY)))
         (PRINTOUT STREAM (SELECTQ (fetch EQUIV of ENTRY)
                                   (T "==")
                                   (NIL "  ")
                                   (CONCAT (SELECTQ (CAR (fetch EQUIV of ENTRY))
                                                    (CR 'C)
                                                    (LF 'L)
                                                    (CRLF 2)
                                                    "x")
                                           (SELECTQ (CADR (fetch EQUIV of ENTRY))
                                                    (CR 'C)
                                                    (LF 'L)
                                                    (CRLF 2)
                                                    "x")))
                 " ")
         (CL:WHEN INFO1
             (PRINTOUT STREAM (SUBSTRING (fetch (CDINFO FULLNAME) OF INFO1)
                                         (ADD1 NCHARSDIR1)
                                         NIL
                                         (CONSTANT (CONCAT)))
                       " ")
             (CL:WHEN PRINTAUTHOR
                 (PRINTOUT STREAM "(" (fetch (CDINFO AUTHOR) OF INFO1)
                           ") "))
             (PRINTOUT STREAM .FR (IDIFFERENCE DATE1POS 2)
                       (fetch (CDINFO LENGTH) OF INFO1)
                       "  "
                       (fetch DATE of INFO1)))
         (PRINTOUT STREAM .TAB0 ENDDATE1 "  " (SELECTQ (fetch DATEREL of ENTRY)
                                                       (< "< ")
                                                       (> " >")
                                                       (* (CL:IF INFO1
                                                                 " *"
                                                                 "* "))
                                                       (SHOULDNT))
                 "  ")
         (CL:WHEN INFO2
             (PRINTOUT STREAM (fetch DATE of INFO2)
                       "  "
                       (SUBSTRING (fetch (CDINFO FULLNAME) OF INFO2)
                                  (ADD1 NCHARSDIR2)
                                  NIL
                                  (CONSTANT (CONCAT)))
                       " ")
             (CL:WHEN PRINTAUTHOR
                 (PRINTOUT STREAM "(" (fetch (CDINFO AUTHOR) OF INFO2)
                           ") "))
             (PRINTOUT STREAM .FR LENGTH2END (fetch (CDINFO LENGTH) OF INFO2))
             (PRINTOUT STREAM " "))                                     ; A little margin in the window
         (TERPRI STREAM])
```

(CDPRINT.MAXWIDTHS

```
[LAMBDA (CDVALUE)                                                       ; Edited 30-Nov-2021 13:51 by rmk:
    ;; This computes the maximum widths needed for a printer to get all the entry-columns lined up.
    ;; The FULLNAME field of INFOs includes the full directory.  The caller is responsible for discounting the lengths of the common directory prefixes.
    ;;
    (LET ((CDENTRIES (CL:IF (STRINGP (FETCH CDDIR2 OF CDVALUE))
                            (FETCH CDENTRIES OF CDVALUE)
                            CDVALUE)))
         (CL:WHEN CDENTRIES
             [LIST (CREATE CDMAXNCHARS
                           NCFULLNAME _ (FOR CD IN CDENTRIES
                                           LARGEST (NCHARS (OR (FETCH (CDINFO FULLNAME)
                                                                      OF (FETCH (CDENTRY INFO1) OF CD))
                                                               ""))
                                           FINALLY (RETURN (OR $$EXTREME 0)))
                           NCLENGTH _ (FOR CD IN CDENTRIES LARGEST (NCHARS (OR (FETCH (CDINFO LENGTH)
                                                                                      OF (FETCH (CDENTRY INFO1)
                                                                                                OF CD))
                                                                               ""))
                                           FINALLY (RETURN (OR $$EXTREME 0)))
                           NCAUTHOR _ (FOR CD IN CDENTRIES LARGEST (NCHARS (OR (FETCH (CDINFO AUTHOR)
                                                                                      OF (FETCH (CDENTRY INFO1)
```

```
                                                                    OF CD))
                                                                ""))
                                        FINALLY (RETURN (OR $$EXTREME 0)))
                                NCTYPE _ (FOR CD IN CDENTRIES LARGEST (NCHARS (OR (FETCH (CDINFO TYPE)
                                                                    OF (FETCH (CDENTRY INFO1)
                                                                        OF CD))
                                                                ""))
                                        FINALLY (RETURN (OR $$EXTREME 0)))
                                NCDIR _ (NCHARS (FETCH (CDVALUE CDDIR1) OF CDVALUE)))
                        (CREATE CDMAXNCHARS
                                NCFULLNAME _ (FOR CD IN CDENTRIES
                                        LARGEST (NCHARS (OR (FETCH (CDINFO FULLNAME)
                                                                OF (FETCH (CDENTRY INFO2) OF CD))
                                                            ""))
                                        FINALLY (RETURN (OR $$EXTREME 0)))
                                NCLENGTH _ (FOR CD IN CDENTRIES LARGEST (NCHARS (OR (FETCH (CDINFO LENGTH)
                                                                    OF (FETCH (CDENTRY INFO2)
                                                                        OF CD))
                                                                ""))
                                        FINALLY (RETURN (OR $$EXTREME 0)))
                                NCAUTHOR _ (FOR CD IN CDENTRIES LARGEST (NCHARS (OR (FETCH (CDINFO AUTHOR)
                                                                    OF (FETCH (CDENTRY INFO2)
                                                                        OF CD))
                                                                ""))
                                        FINALLY (RETURN (OR $$EXTREME 0)))
                                NCTYPE _ (FOR CD IN CDENTRIES LARGEST (NCHARS (OR (FETCH (CDINFO TYPE)
                                                                    OF (FETCH (CDENTRY INFO2)
                                                                        OF CD))
                                                                ""))
                                        FINALLY (RETURN (OR $$EXTREME 0)))
                                NCDIR _ (NCHARS (FETCH (CDVALUE CDDIR2) OF CDVALUE])])
```

(CDPRINT.COLHEADERS

```
[LAMBDA (STREAM COLHEADINGS ENDDATE1 COL1WIDTH COL2START COL2WIDTH)
                                                                    ; Edited 16-Jul-2022 10:38 by rmk
                                                                    ; Edited 30-Nov-2021 14:47 by rmk:

    ;; If column headers are provided, center them over the columns

    (CL:WHEN (LISTP COLHEADINGS)
        (LET (HEADING)
             (CL:WHEN (SETQ HEADING (CAR COLHEADINGS))
                 (CL:WHEN (IGREATERP (NCHARS HEADING)
                                     COL1WIDTH)                 ; Truncate to column width
                     (SETQ HEADING (SUBSTRING HEADING 1 COL1WIDTH)))
                 (TAB (DIFFERENCE ENDDATE1 COL1WIDTH)
                      0 STREAM)
                 (FLUSHRIGHT ENDDATE1 HEADING 0 NIL T STREAM))
             (CL:WHEN [SETQ HEADING (CAR (LISTP (CDR COLHEADINGS]
                 (CL:WHEN (IGREATERP (NCHARS HEADING)
                                     COL2WIDTH)
                     (SETQ HEADING (SUBSTRING HEADING 1 COL2WIDTH)))
                 (TAB COL2START 0 STREAM)
                 (FLUSHRIGHT (PLUS COL2START COL2WIDTH)
                        HEADING 0 NIL T STREAM))
             (TERPRI STREAM)))])
```

(CDPRINT.COLUMNS

```
[LAMBDA (CDVALUE COLHEADINGS PRINTAUTHOR)                       ; Edited 20-Jul-2022 08:53 by rmk
                                                                ; Edited 16-Jul-2022 10:40 by rmk
                                                                ; Edited 30-Nov-2021 14:03 by rmk:

    ;; Compute the column locations for CDPRINT.LINE

    ;; Even though the longest length and author might not go with the longest file name, it is a reasonable approximation to assume that in fact the
    ;; longest filename did have the longest length.  Lengths differ by just a few characters, and a long length with a short filename might balance out.
    ;; If the long file did have a long length, then it would all be exact.

    (SETQ CDVALUE (CD.INSURECDVALUE CDVALUE))
    (LET (INFO1 DATE1POS ENDDATE1 (COL1WIDTH 10)
                (COL2WIDTH 10)
                (DATERELWIDTH 6)
                (MAXWIDTHS1 (FETCH (CDVALUE CDMAXNC1) OF CDVALUE))
                (MAXWIDTHS2 (FETCH (CDVALUE CDMAXNC2) OF CDVALUE))
                (MAXAUTHOR1 0)
                (MAXAUTHOR2 0)
                [DATEWIDTH (CONSTANT (NCHARS (DATE]
                MAXNAME1 MAXNAME2 (EQUIV 4)
                COL2START LENGTH2END)

        ;; DATE1POS is the position of the first character of INFO1's date, used for tabbing.  We have to measure the filename, date, size, and
        ;; author if desired

        (if (fetch CDENTRIES of CDVALUE)
            then ;; Compute the column locations

                ;; Even though the longest length and author might not go with the longest file name, it is a reasonable approximation to assume
                ;; that in fact the longest filename did have the longest length.  Lengths differ by just a few characters, and a long length with a
                ;; short filename might balance out.  If the long file did have a long length, then it would all be exact.
```

```
                [SETQ MAXNAME1 (IF (IGREATERP (fetch NCFULLNAME of MAXWIDTHS1)
                                              0)
                                   THEN (IDIFFERENCE (fetch NCFULLNAME of MAXWIDTHS1)
                                                     (fetch NCDIR OF MAXWIDTHS1))
                                   ELSE                    ; Nothing in column 1, space out a bit
                                        (IMAX 20 (NCHARS (CAR (LISTP COLHEADINGS]
                [SETQ MAXNAME2 (IF (IGREATERP (fetch NCFULLNAME of MAXWIDTHS2)
                                              0)
                                   THEN (IDIFFERENCE (fetch NCFULLNAME of MAXWIDTHS2)
                                                     (fetch NCDIR OF MAXWIDTHS2))
                                   ELSE (IMAX 20 (NCHARS (CAR (LISTP COLHEADINGS]
                (CL:WHEN PRINTAUTHOR
                    ;; MAXAUTHOR includes its own suffixspace
                    [SETQ MAXAUTHOR1 (IPLUS (CONSTANT (NCHARS "("))
                                            (fetch NCAUTHOR of MAXWIDTHS1)
                                            (CONSTANT (NCHARS ") "]
                    [SETQ MAXAUTHOR2 (IPLUS (CONSTANT (NCHARS (NCHARS "(")))
                                            (fetch NCAUTHOR of MAXWIDTHS2)
                                            (CONSTANT (NCHARS ") "])
                (SETQ COL1WIDTH (IPLUS MAXNAME1 1 MAXAUTHOR1 (fetch NCLENGTH of MAXWIDTHS1)
                                       2 DATEWIDTH))
                (SETQ DATE1POS (IPLUS EQUIV (IDIFFERENCE COL1WIDTH DATEWIDTH)))
                (SETQ ENDDATE1 (IPLUS EQUIV COL1WIDTH))
                (SETQ COL2WIDTH (IPLUS DATEWIDTH 2 MAXNAME2 1 MAXAUTHOR2 (fetch NCLENGTH of MAXWIDTHS2)))
                ;; If column headers are provided, center them over the columns. But don't expand the column, the headers will be truncated.
                                                        (* (CL:WHEN (CAR (LISTP COLHEADINGS))
                                                        (SETQ COL1WIDTH (IMAX 10 (NCHARS
                                                        (CAR COLHEADINGS)) COL1WIDTH))))
                (SETQ COL2START (PLUS EQUIV COL1WIDTH DATERELWIDTH))
                                                        (* (CL:WHEN (CAR (LISTP (CDR COLHEADINGS)))
                                                        (SETQ COL2WIDTH (IMAX 10 (NCHARS
                                                        (CADR COLHEADINGS)) COL2WIDTH))))
                (SETQ LENGTH2END (IPLUS COL2START COL2WIDTH))
                (LIST DATE1POS ENDDATE1 COL1WIDTH COL2WIDTH COL2START LENGTH2END])
```

(CDTEDIT

```
  [LAMBDA (CDVALUE TITLE COLHEADINGS PRINTAUTHOR)           ; Edited  5-Nov-2021 16:44 by rmk:
                                                            ; Edited 31-Oct-2021 11:02 by rmk:
    ;; CDPRINT to a read-only TEDIT file.
    (LET ((TSTREAM (OPENTEXTSTREAM)))
         (DSPFONT DEFAULTFONT TSTREAM)
         (CDPRINT CDVALUE TSTREAM COLHEADINGS PRINTAUTHOR)
         (TERPRI TSTREAM)
         (TEDIT TSTREAM NIL NIL `(READONLY T WINDOWTYPE CDTEDIT TITLE ,(OR TITLE "Compare directories"])

)

(DEFINEQ
```

(CDMAP

```
  [LAMBDA (CDVALUE FN)                                      ; Edited  5-Nov-2021 16:46 by rmk:
                                                            ; Edited  6-Sep-2020 15:58 by rmk:
    (CL:UNLESS CDVALUE
        (PRINTOUT T T "Note:  Using LASTCDVALUE" T T)
        (SETQ CDVALUE LASTCDVALUE))
    (FOR CDE MATCHNAME INFO1 DATEREL INFO2 EQUIV IN (FETCH CDENTRIES OF CDVALUE)
       DECLARE (SPECVARS MATCHNAME INFO1 DATEREL INFO2 EQUIV) EACHTIME (SETQ MATCHNAME (FETCH MATCHNAME
                                                                                           OF CDE))
                                                                       (SETQ INFO1 (FETCH INFO1 OF CDE))
                                                                       (SETQ DATEREL (FETCH DATEREL OF CDE))
                                                                       (SETQ INFO2 (FETCH INFO2 OF CDE))
                                                                       (SETQ EQUIV (FETCH EQUIV OF CDE))
       DO (APPLY* FN CDE])
```

(CDENTRY

```
  [LAMBDA (MATCHNAME CDVALUE)                               ; Edited  5-Nov-2021 16:47 by rmk:
                                                            ; Edited  5-Sep-2020 21:09 by rmk:
    (ASSOC MATCHNAME (FETCH CDENTRIES OF (OR CDVALUE LASTCDVALUE])
```

(CDSUBSET

```
  [LAMBDA (CDVALUE FN)                                      ; Edited  4-Dec-2021 09:08 by rmk
                                                            ; Edited 30-Nov-2021 11:01 by rmk:
                                                            ; Edited  5-Nov-2021 16:56 by rmk:
                                                            ; Edited 15-Sep-2020 13:49 by rmk:
    (SETQ CDVALUE (CD.INSURECDVALUE CDVALUE))
    (CD.UPDATEWIDTHS (CREATE CDVALUE USING CDVALUE CDENTRIES _ (FOR CDE MATCHNAME INFO1 DATEREL INFO2 EQUIV
                                                                  IN (FETCH CDENTRIES OF CDVALUE)
                                                                  DECLARE (SPECVARS MATCHNAME INFO1 DATEREL
                                                                                    INFO2 EQUIV)
                                                                  EACHTIME (SETQ MATCHNAME
                                                                                 (FETCH MATCHNAME OF CDE))
```

```
                                        (SETQ INFO1 (FETCH INFO1 OF CDE))
                                        (SETQ DATEREL (FETCH DATEREL
                                                             OF CDE))
                                        (SETQ INFO2 (FETCH INFO2 OF CDE))
                                        (SETQ EQUIV (FETCH EQUIV OF CDE))
                                  WHEN (APPLY* FN CDE) COLLECT CDE])
```

(CDMERGE

```
[LAMBDA (CDVALUES)                                          ; Edited  5-Apr-2023 10:10 by rmk
                                                            ; Edited 24-Jan-2022 17:01 by rmk
```

;; This merges a collection of CDVALUES on different directories into a single CDVALUE with the union of the CDENTRIES, provided that they
;; have the same selection criteria. The merged directories will be the minimal common prefix of all of the entries on each side, and the residual of
;; the directory will be packed onto all the names.

```
(IF (CDR CDVALUES)
    THEN
    [LET (CDSELECTS)
         ;; Group by selects
         (FOR CDV TMP IN CDVALUES
            DO (PUSH [CDR (OR (SASSOC (FETCH CDSELECT OF CDV)
                                      CDSELECTS)
                              (CAR (PUSH CDSELECTS (CONS (FETCH CDSELECT OF CDV]
                       CDV))
         ;; For each group, find the longest common directory prefixes
         (FOR CDS IDATE DIR1 DIR2 MERGEDENTRIES IN CDSELECTS
            COLLECT (SETQ DIR1 (FETCH CDDIR1 OF (CADR CDS)))
                    (SETQ DIR2 (FETCH CDDIR2 OF (CADR CDS)))
                    [SETQ IDATE (IDATE (FETCH CDCOMPAREDATE OF (CADR CDS]
                    ;; Calculate the common directory prefixes and latest date
                    [FOR CDV IN (CDDR CDS) DO (SETQ DIR1 (CDMERGE.COMMON DIR1 (FETCH CDDIR1 OF CDV)))
                                              (SETQ DIR2 (CDMERGE.COMMON DIR2 (FETCH CDDIR2 OF CDV)))
                                              (CL:WHEN (IGREATERP IDATE (IDATE (FETCH CDCOMPAREDATE
                                                                                   OF CDV)))
                                                  (SETQ IDATE (IDATE (FETCH CDCOMPAREDATE OF CDV))))]
                    ;; Merge the CDENTRIES with matchnames pulled back so that subdirectories show up
                    (SETQ MERGEDENTRIES
                     (SORT [FOR CDV (NC1 _ (ADD1 (NCHARS DIR1)))
                                (NC2 _ (ADD1 (NCHARS DIR2))) IN (CDR CDS)
                           JOIN (FOR CDE IN (FETCH CDENTRIES OF CDV)
                                   COLLECT (CREATE CDENTRY
                                                   USING CDE MATCHNAME _
                                                         (IF (FETCH INFO1 OF CDE)
                                                             THEN (MATCHNAME (FETCH (CDINFO FULLNAME)
                                                                                    OF (FETCH INFO1 OF CDE))
                                                                             NC1)
                                                             ELSE (MATCHNAME (FETCH (CDINFO FULLNAME)
                                                                                    OF (FETCH INFO2 OF CDE))
                                                                             NC2]
                           (FUNCTION CD.SORT)))
                    (CD.UPDATEWIDTHS (CREATE CDVALUE
                                             CDDIR1 _ DIR1
                                             CDDIR2 _ DIR2
                                             CDCOMPAREDATE _ (GDATE IDATE)
                                             CDSELECT _ (CAR CDS)
                                             CDENTRIES _ MERGEDENTRIES]
    ELSE CDVALUES])
```

(CDMERGE.COMMON

```
[LAMBDA (DIRX DIRY)                                         ; Edited 24-Jan-2022 16:40 by rmk
```

;; Returns the longest common prefix of DIRX and DIRY, collapsing brackets, slashes, and case

```
(FOR I CX CY (LASTDIRPOS _ 1) FROM 1 EACHTIME (SETQ CX (NTHCHARCODE DIRX I))
                                              (SETQ CY (NTHCHARCODE DIRY I))
                                              (CL:WHEN (MEMB CX (CHARCODE (< > /)))
                                                  (SETQ CX (CHARCODE /)))
                                              (CL:WHEN (MEMB CY (CHARCODE (< > /)))
                                                  (SETQ CY (CHARCODE /)))
                                              (CL:WHEN (AND (EQ CX (CHARCODE /))
                                                            (EQ CY (CHARCODE /)))
                                                  (SETQ LASTDIRPOS I))
    UNLESS [AND CX CY (OR (EQ CX CY)
                          (EQ (L-CASECODE CX)
                              (L-CASECODE CY]
    DO (RETURN (CL:IF (EQ I 1)
                      ""
                      (SUBSTRING DIRX 1 LASTDIRPOS))])
```

(CD.SORT

```
[LAMBDA (ENTRY1 ENTRY2)                                     ; Edited  5-Apr-2023 10:15 by rmk
```

;; Groups same file with different extensions together. FOO and FOO.LCOM together, even if FOO-FUM exists (hyphen comes before period).

```
    (LET ((M1 (FETCH MATCHNAME OF ENTRY1))
          (M2 (FETCH MATCHNAME OF ENTRY2))
          ORDER)
         (CL:IF [EQ 'EQUAL (SETQ ORDER (ALPHORDER (PACKFILENAME 'EXTENSION NIL 'BODY M1)
                                                  (PACKFILENAME 'EXTENSION NIL 'BODY M2]
             (ALPHORDER M1 M2)
             ORDER)])

)

(DEFINEQ
```

(BINCOMP

```
  [LAMBDA (FILE1 FILE2 EOLDIFFOK EOL1 EOL2)                    ; Edited 13-Oct-2020 08:53 by rmk:
    ;; Returns T if FILE1 and FILE2 are byte-equivalent.  Returns EOLDIFF if they are byte equivalent except for CR/LF/CRLF exchanges.
    ;; If EOLDIFFOK, return indicates that the files are the same except for EOL mappings.  If EOL1 and EOL2 are not provided, they are computed
    ;; here.
    (IF (IEQP (GETFILEINFO FILE1 'LENGTH)
              (GETFILEINFO FILE2 'LENGTH))
        THEN [CL:WITH-OPEN-FILE (STREAM1 FILE1 :DIRECTION :INPUT)
                    (CL:WITH-OPEN-FILE (STREAM2 FILE2 :DIRECTION :INPUT)
                           (SETFILEINFO STREAM1 'ENDOFSTREAMOP (FUNCTION NILL))
                           ;; Simpler code to recompute eol's even if provided
                           (BIND B1 B2 EOL1 EOL2 EOLDIFF WHILE (SETQ B1 (\BIN STREAM1))
                              UNLESS (EQ B1 (SETQ B2 (\BIN STREAM2)))
                              DO (CL:UNLESS (AND EOLDIFFOK (SELCHARQ B1
                                                                    (CR (CL:WHEN (EQ EOL1 'LF)
                                                                                 (RETURN NIL))
                                                                        (SETQ EOL1 'CR)
                                                                        (SETQ EOL2 'LF)
                                                                        (EQ B2 (CHARCODE LF)))
                                                                    (LF (CL:WHEN (EQ EOL1 'CR)
                                                                                 (RETURN NIL))
                                                                        (SETQ EOL1 'LF)
                                                                        (SETQ EOL2 'CR)
                                                                        (EQ B2 (CHARCODE CR)))
                                                                    NIL))
                                                 (RETURN NIL))
                                            (CL:UNLESS EOLDIFF
                                                (SETQ EOLDIFF (LIST EOL1 EOL2)))
                              FINALLY (RETURN (OR EOLDIFF T]
      ELSEIF EOLDIFFOK
        THEN ;; Lengths are different possibly because of CRLF to CR/LF substitutions.
             ;; More complex code could detect the EOLTYPE incrementally without separate passes, but ...
             (CL:UNLESS EOL1
                 (SETQ EOL1 (EOLTYPE FILE1)))
             (CL:UNLESS EOL2
                 (SETQ EOL2 (EOLTYPE FILE2)))
             (CL:WHEN (IF [AND (EQ EOL1 'CRLF)
                               (MEMB EOL2 '(LF CR]
                          ELSEIF [AND (EQ EOL2 'CRLF)
                                      (MEMB EOL1 '(LF CR]
                            THEN (SWAP FILE1 FILE2))
                 ;; FILE1 is now CRLF, FILE2 is not.  If FILE1 isn't longer, it can't have a CRLF that corresponds to a CR or LF.
                 (CL:WHEN (IGREATERP (GETFILEINFO FILE1 'LENGTH)
                                     (GETFILEINFO FILE2 'LENGTH))
                     [CL:WITH-OPEN-FILE (STREAM1 FILE1 :DIRECTION :INPUT)
                            (CL:WITH-OPEN-FILE (STREAM2 FILE2 :DIRECTION :INPUT)
                                   (SETFILEINFO STREAM1 'ENDOFSTREAMOP (FUNCTION NILL))
                                   (BIND B1 B2 EOLDIFF WHILE (SETQ B1 (\BIN STREAM1))
                                      UNLESS (EQ B1 (SETQ B2 (\BIN STREAM2)))
                                      DO (CL:UNLESS [AND (EQ (CHARCODE CR)
                                                             B1)
                                                         (EQ (CHARCODE LF)
                                                             (\BIN STREAM1))
                                                         (MEMB B2 (CHARCODE (CR LF]
                                                (RETURN NIL))
                                            (CL:UNLESS EOLDIFF
                                                (SETQ EOLDIFF (LIST EOL1 EOL2)))
                                      FINALLY (RETURN (OR EOLDIFF T]))])
```

(EOLTYPE

```
  [LAMBDA (FILE SHOWCONTEXT)
    ;; Edited  4-Jan-2022 15:10 by rmk: Allow FILE to be an already open stream
    ;; Edited 21-Feb-2021 20:34 by rmk:
    ;; Returns the EOLCONVENTION of FILE if it only sees one kind, NIL if it can't decide.
    ;; If SHOWCONTEXT, it is the number of bytes before and after an EOL inconsistency (e.g. seeing CR after having seen LF) that will be displayed
    ;; on the TTY.  The position of the inconsistency will be marked with ##.
```

```
    (SELECTQ SHOWCONTEXT
         (NIL)
         (T (SETQ SHOWCONTEXT 100))
         (CL:UNLESS (FIXP SHOWCONTEXT)
                (ERROR "SHOWCONTEXT must be an integer" SHOWCONTEXT)))
    (RESETLST
        (LET (STREAM)
             [IF (GETSTREAM FILE 'INPUT T)
                 THEN (SETQ STREAM FILE)
                      [RESETSAVE NIL `(PROGN (SETFILEPTR ,STREAM ,(GETFILEPTR STREAM))
                                             (STREAMPROP ,STREAM 'ENDOFSTREAMOP
                                                    ',(STREAMPROP STREAM 'ENDOFSTREAMOP]
                      (SETFILEPTR STREAM 0)
               ELSE (RESETSAVE NIL `(CLOSEF? ,(SETQ STREAM (OPENSTREAM FILE 'INPUT]
             (SETFILEINFO STREAM 'ENDOFSTREAMOP (FUNCTION NILL))
             (BIND EOLTYPE DO (SELCHARQ (OR (\BIN STREAM)
                                            (RETURN EOLTYPE))
                                        (CR (IF (EQ (CHARCODE LF)
                                                    (\PEEKBIN STREAM T))
                                                THEN (\BIN STREAM)
                                                     (IF (MEMB EOLTYPE '(LF CR))
                                                         THEN (CL:UNLESS (EOLTYPE.SHOW SHOWCONTEXT EOLTYPE
                                                                                       'LF STREAM)
                                                                         (RETURN NIL))
                                                         ELSE (SETQ EOLTYPE 'CRLF))
                                              ELSEIF (MEMB EOLTYPE '(LF CRLF))
                                                THEN (CL:UNLESS (EOLTYPE.SHOW SHOWCONTEXT EOLTYPE 'CR STREAM)
                                                                (RETURN NIL))
                                              ELSE (SETQ EOLTYPE 'CR)))
                                        (LF (IF (MEMB EOLTYPE '(CR CRLF))
                                                THEN (CL:UNLESS (EOLTYPE.SHOW SHOWCONTEXT EOLTYPE 'LF STREAM)
                                                                (RETURN NIL))
                                                ELSE (SETQ EOLTYPE 'LF)))
                                        NIL)))))])
```

(EOLTYPE.SHOW

```
  [LAMBDA (SHOWCONTEXT OLDTYPE NEWTYPE STREAM)                    ; Edited 21-Feb-2021 20:20 by rmk:
```

;; Returns T if we should continue

```
    (CL:WHEN SHOWCONTEXT
        (LET ((FILEPOS (GETFILEPTR STREAM)))
             (COPYBYTES STREAM T (IDIFFERENCE FILEPOS SHOWCONTEXT)
                    FILEPOS)
             (PRINTOUT T OLDTYPE "->" NEWTYPE " " FILEPOS T)
             (COPYBYTES STREAM T FILEPOS (IPLUS FILEPOS SHOWCONTEXT))
             (TERPRI T)
             (CL:WHEN (EQ 'Y (ASKUSER NIL NIL "Continue?  "))
                 (PRINTOUT T T "-------" T T)
                 (SETFILEPTR STREAM FILEPOS)
                 T))))])

)

(DECLARE%: EVAL@COMPILE

(RECORD CDMAXNCHARS (NCFULLNAME NCLENGTH NCAUTHOR NCTYPE NCDIR))

(RECORD CDVALUE ((CDDIR1 CDDIR2 CDCOMPAREDATE CDSELECT CDMAXNC1 CDMAXNC2) . CDENTRIES)
       (RECORD CDVALUE (CDPARAMETERS))
       CDMAXNC1 _ (CREATE CDMAXNCHARS)
       CDMAXNC2 _ (CREATE CDMAXNCHARS))

(RECORD CDENTRY (MATCHNAME INFO1 DATEREL INFO2 . EQUIV))

(RECORD CDINFO (FULLNAME DATE LENGTH AUTHOR TYPE EOL))
)
```

;; look for compiled files older than the sources

```
(DEFINEQ
```

(FIND-UNCOMPILED-FILES

```
  [LAMBDA (FILES DFASLMARGIN COMPILEEXTS)                         ; Edited 20-Sep-2020 23:04 by rmk:
                                                                  ; Edited  3-Nov-94 15:17 by jds
```

;; Produces a list of the source files in FILES that have no corresponding compiled file

;; This determines whether there is at least one compiled file. If there are two or more, that's a problem

;; We want the most recent version only

;; Source files have a 2-element created-as with a non-NIL date

```
    (SETQ FILES (FOR F IN (OR (LISTP FILES)
                              (FILDIR FILES))
                   UNLESS (MEMB (SETQ F (PACKFILENAME 'VERSION NIL 'BODY F))
                                $$VAL)
                   COLLECT F))
```

```
(FOR F SCREATION FILES IN FILES WHEN (AND (CADR (SETQ SCREATION (CREATED-AS F)))
                                          (NOT (CDDR SCREATION)))
   WHEN [SETQ FILES (FOR CEXT CF IN (OR COMPILEEXTS *COMPILED-EXTENSIONS*)
                       WHEN (SETQ CF (INFILEP (PACKFILENAME 'EXTENSION CEXT 'VERSION NIL 'BODY F)))
                       COLLECT (CL:WHEN (SOURCE-FOR-COMPILED-P SCREATION CF DFASLMARGIN)
                                        (RETURN NIL))
                             CF
                       FINALLY
                             ;; If we found some compiled files, they weren't on this source.  If there weren't any compiled files to
                             ;; check, maybe there weren't any functions.
                             ;; NLSETQ because we don't want to stop if there is an error, typically from a package problem
                             (RETURN (OR $$VAL (LET [(FCOMS (CAR (NLSETQ (GETDEF (FILECOMS F)
                                                                                  'VARS F]
                                                                    (IF (NULL FCOMS)
                                                                        THEN ;; GETDEF caused an error.  Maybe a package problem.
                                                                             (AND NIL 'NOCOMMANDS)
                                                                      ELSEIF (INFILECOMS? NIL '(FUNCTIONS FNS)
                                                                                     FCOMS)
                                                                        THEN T]
   COLLECT (CONS F (SELECTQ FILES
                         (T NIL)
                         (NOCOMMANDS (CONS "No commands"))
                         (FOR CF IN FILES COLLECT
                                             ;; Positive means that compiled is later than source, normal order but maybe by
                                             ;; too much.
                                             ;; Negative means that compiled came before source.  Odd
                                             (LIST CF (COMPILE-SOURCE-DATE-DIFF CF SCREATION])
```

(FIND-UNSOURCED-FILES

```
[LAMBDA (FILES DFASLMARGIN COMPILEEXTS)                    ; Edited 25-Apr-2022 08:43 by rmk
                                                           ; Edited 15-Sep-2020 15:32 by rmk:
                                                           ; Edited  3-Nov-94 15:17 by jds
```

;; Produces a list of compiled FILES for which no source file can be found in the same directory.

;; The source date in at least one DFASL was off by a second, maybe some sort of IDATE rounding? So, give a margin.

;; We want the most recent version only. Check CREATED-AS to make sure it really is a compiled file.

;; Sort to get lcoms and dfasls next to each other.

```
(LET (CCREATEDS)
     (SETQ CCREATEDS (FOR CEXT FOUND CCREATED INSIDE (OR COMPILEEXTS *COMPILED-EXTENSIONS*)
                        JOIN (FOR CF IN [OR (LISTP FILES)
                                            (FILDIR (PACKFILENAME.STRING 'EXTENSION CEXT 'VERSION ""
                                                                         'BODY
                                                                         '*]
                                  WHEN (CDDR (SETQ CCREATED (CREATED-AS CF))) UNLESS (MEMBER CCREATED $$VAL
                                                                                             )
                                  COLLECT CCREATED)))
       ;; CCREATEDS is now a list of CREATED-AS items
       (FOR CC SF IN CCREATEDS UNLESS (AND [SETQ SF (INFILEP (PACKFILENAME.STRING 'EXTENSION NIL 'VERSION NIL
                                                                                  'BODY
                                                                                  (CAR CC]
                                           (SOURCE-FOR-COMPILED-P (SETQ SF (CREATED-AS SF))
                                                  CC DFASLMARGIN))
          COLLECT [LIST (CAR CC)
                        (AND SF (LIST (CAR SF)
                                      (ROUND (COMPILE-SOURCE-DATE-DIFF CC SF]
          FINALLY (RETURN (SORT $$VAL (FUNCTION (LAMBDA (CF1 CF2)
                                                  (ALPHORDER (FILENAMEFIELD.STRING (CAR CF1)
                                                                                   'NAME)
                                                             (FILENAMEFIELD.STRING (CAR CF2)
                                                                                   'NAME])
```

(FIND-SOURCE-FILES

```
[LAMBDA (CFILES SDIRS DFASLMARGIN)                         ; Edited 25-Apr-2022 08:43 by rmk
                                                           ; Edited  9-Sep-2020 12:26 by rmk:
```

;; Returns (CFILE . SFILES) pairs where CFILE is a Lisp compiled file in CFILES SFILES is a list of source files in SDIRS that CFILE was compiled ;; on.

;; This suggests that one of CFILES should be copied to the SFILE directory.

```
(SETQ SDIRS (FOR SD INSIDE (OR SDIRS T) COLLECT (DIRECTORYNAME SD)))
(SORT (FOR CF SFILES CNAME CCREATED IN (OR (LISTP CFILES)
                                           (FILDIR CFILES))
         WHEN (AND (SETQ CNAME (INFILEP CF))
                   (CDDR (SETQ CCREATED (CREATED-AS CF)))
                   (SETQ SFILES (FOR SD SF IN SDIRS WHEN (AND (SETQ SF (INFILEP (PACKFILENAME.STRING
                                                                                 'NAME
                                                                                 (FILENAMEFIELD.STRING
                                                                                  CF
                                                                                  'NAME)
```

```
                                                                    'BODY SD)))
                                                          (SOURCE-FOR-COMPILED-P SF CCREATED
                                                              DFASLMARGIN))
                                        COLLECT SF)))
          COLLECT (CONS CNAME SFILES))
       (FUNCTION (LAMBDA (P1 P2)
                   (ALPHORDER (FILENAMEFIELD.STRING (CAR P1))
                          (FILENAMEFIELD.STRING (CAR P2])
```

(FIND-COMPILED-FILES

```
[LAMBDA (SFILES CDIRS DFASLMARGIN)                          ; Edited 25-Apr-2022 08:44 by rmk
                                                            ; Edited  9-Sep-2020 12:26 by rmk:
```

;; Returns (SFILE . CFILES) pairs where SFILE is a Lisp source file in SFILES CFILES is a list of compiled files in CDIRS that were compiled on ;; SFILE.

;; FILEDATE is true for source files and compiled files

;; This suggests that one of CFILES should be copied to the SFILE directory.

```
(SETQ CDIRS (FOR CD INSIDE (OR CDIRS T) COLLECT (DIRECTORYNAME CD)))
(SORT (FOR SF CFILES SNAME SCREATED IN (OR (LISTP SFILES)
                                           (FILDIR SFILES))
         WHEN [AND (SETQ SNAME (INFILEP SF))
                   (SETQ SCREATED (CREATED-AS SF))
                   (NOT (CDDR SCREATED))
                   (SETQ CFILES (FOR CEXT (ROOT _ (FILENAMEFILEFIELD.STRING SNAME 'NAME)) IN
                                                                                *COMPILED-EXTENSIONS*
                                   JOIN (FOR CD CF IN CDIRS
                                           WHEN (AND (SETQ CF (INFILEP (PACKFILENAME.STRING 'NAME ROOT
                                                                        'EXTENSION CEXT 'BODY CD)))
                                                     (SOURCE-FOR-COMPILED-P SCREATED CF DFASLMARGIN))
                                           COLLECT CF]
         COLLECT (CONS SNAME CFILES))
       (FUNCTION (LAMBDA (P1 P2)
                   (ALPHORDER (FILENAMEFIELD.STRING (CAR P1))
                          (FILENAMEFIELD.STRING (CAR P2])
```

(FIND-UNLOADED-FILES

```
[LAMBDA (FILES)                                             ; Edited 25-Apr-2022 08:49 by rmk
                                                            ; Edited  9-Sep-2020 19:35 by rmk:
```

;; Returns the files in FILES that don't have FILECREATED properties and presumably are therefore not loaded in the current sysout.

```
(FOR F IN (OR (LISTP FILES)
              (FILDIR FILES))
   WHEN (AND (SETQ F (INFILEP (CL:IF (LISTP F)
                                     (CAR F)
                                    F)))
             (FILEDATE F))
   UNLESS (GETP (FILENAMEFIELD F 'NAME)
                'FILEDATES)
   COLLECT F])
```

(FIND-LOADED-FILES

```
[LAMBDA (ROOTFILENAMES)                                     ; Edited 25-Apr-2022 09:04 by rmk
                                                            ; Edited 19-Sep-2020 07:20 by rmk:
```

```
(FOR RN INSIDE ROOTFILENAMES WHEN (GETP RN 'FILEDATES)
   COLLECT (CONS RN (FOR F IN LOADEDFILELST WHEN (EQ RN (FILENAMEFIELD F 'NAME)) COLLECT F])
```

(FIND-MULTICOMPILED-FILES

```
[LAMBDA (FILES SHOWINFO)                                    ; Edited 25-Apr-2022 09:07 by rmk
                                                            ; Edited 20-Sep-2020 20:57 by rmk:
```

;; Returns a list of names for files in FILES that have multiple compilations

```
(LET (SFILES)
     (FOR F EXT NAME IN (OR (LISTP FILES)
                            (FILDIR FILES))
        WHEN (MEMB (SETQ EXT (FILENAMEFIELD F 'EXTENSION))
                   *COMPILED-EXTENSIONS*)
        DO (SETQ NAME (FILENAMEFIELD F 'NAME))
           ;; PUSHNEW because we haven't filtered out versions
           (PUSHNEW [CDR (OR (ASSOC NAME SFILES)
                             (CAR (PUSH SFILES (CONS NAME]
                    EXT))
     (FOR S IN SFILES WHEN (CDDR S)
        COLLECT (IF SHOWINFO
                    THEN `[,(CAR S)
                           ,(CADAR (FIND-LOADED-FILES (CAR S)))
                           ,(CREATED-AS (CAR S))
                           ,@(FOR EXT IN (SORT (CDR S)) COLLECT (CREATED-AS (PACKFILENAME.STRING
                                                                             'EXTENSION EXT 'BODY
                                                                             (CAR S]
                    ELSE (CAR S])
```

```
)

(DEFINEQ
```

(CREATED-AS

```
  [LAMBDA (FILE)                                          ; Edited 20-Sep-2020 23:06 by rmk:
    ;; For lisp source files, returns (filecreatename filecreateddate)
    ;; For lisp compiled files, returns (cfilename cfiledate sfilecreatename sfilecreateddate)
    ;; For other files, (fullfilename NIL)
    ;; The cfilename is just the current directory name for DFASLs.
    ;; So:  (CADR value) is non-NIL for Lisp files.  Of those, (CDDR value) is non-NIL for compiled files.
    ;; We disable the package delimiter because the atoms in changes may have a packages that we don't know.
    (CL:WITH-OPEN-FILE
     (STREAM FILE :DIRECTION :INPUT)
     (LET
      (FILEDATE FILENAME SOURCEDATE SOURCENAME LINE POS)
      [IF (EQ (CHARCODE %()
              (SKIPSEPRCODES STREAM))
          THEN                                            ; Managed source or LCOM
               (RESETLST
                   [LET (FORM SFORM (RDTBL (FIND-READTABLE "OLD-INTERLISP-FILE")))
                        (SETQ POS (GETFILEPTR STREAM))
                        (READCCODE STREAM)
                        (IF (EQ 'DEFINE-FILE-INFO (RATOM STREAM RDTBL))
                            THEN ;; Reading is package-safe
                                 (SETFILEPTR STREAM POS)
                                 (SETQ FORM (READ STREAM RDTBL))
                                 (SETQ RDTBL (FIND-READTABLE (LISTGET (CDR FORM)
                                                                      :READTABLE)))
                          ELSE (SETFILEPTR STREAM POS))
                        (CL:WHEN (EQ 'PACKAGEDELIM (GETSYNTAX '%: RDTBL))
                            [RESETSAVE (SETSYNTAX '%: 'OTHER RDTBL)
                                   `(SETSYNTAX %: PACKAGEDELIM ,RDTBL])
                        ;; One way or the other, we're ready for the filecreated
                        (CL:WHEN (EQ (CHARCODE %()
                                     (SKIPSEPRCODES STREAM))
                            (SETQ FORM (READ STREAM RDTBL))
                            (CL:WHEN (MEMB (U-CASE (CAR FORM))
                                           '(FILECREATED IL%:FILECREATED))
                                ;; IL%:FILECREATED because we screwed the readtable.
                                (IF [STREQUAL "compiled on " (CAR (LISTP (CADDR FORM]
                                    THEN                  ; LCOM, get source info
                                         (IF [AND (EQ (CHARCODE %()
                                                      (SKIPSEPRCODES STREAM))
                                                  (MEMB [U-CASE (CAR (SETQ SFORM (READ STREAM RDTBL]
                                                        '(FILECREATED IL%:FILECREATED]
                                             THEN (SETQ FILENAME (FULLNAME STREAM))
                                                  (SETQ FILEDATE (CADR FORM))
                                                  (SETQ SOURCENAME (CADDR SFORM))
                                                  (SETQ SOURCEDATE (CADR SFORM))
                                           ELSE (SETQ FILENAME (FULLNAME STREAM))
                                                (SETQ FILEDATE (CADR FORM)))
                                      ELSE (SETQ FILENAME (CADDR FORM))
                                           (SETQ FILEDATE (CADR FORM)))))])
        ELSEIF (SETQ POS (STRPOS "XCL Compiler output for source file " (SETQ LINE (CL:READ-LINE STREAM))
                                 1 NIL NIL T))
          THEN                                            ; DFASL compiled?
               (SETQ SOURCENAME (SUBATOM LINE POS))
               (CL:WHEN (SETQ POS (STRPOS "Source file created " (SETQ LINE (CL:READ-LINE STREAM))
                                          1 NIL NIL T))
                   [SETQ SOURCEDATE (GDATE (IDATE (SUBSTRING LINE POS]
                   (CL:WHEN (SETQ POS (STRPOS "FASL file created " (SETQ LINE (CL:READ-LINE STREAM))
                                              1 NIL NIL T))
                       [SETQ FILEDATE (GDATE (IDATE (SUBSTRING LINE POS]))]
      ;; Revert filenames to Interlisp package if needed:
      (CL:WHEN (STRPOS "IL:" FILENAME)
          (SETQ FILENAME (SUBATOM FILENAME 4)))
      (CL:WHEN (STRPOS "IL:" SOURCENAME)
          (SETQ SOURCENAME (MKATOM SOURCENAME 4)))
      ;; Return DATE NIL if file is not a Lisp file
      `(,(OR FILENAME (FULLNAME STREAM))
        ,(AND FILEDATE (GDATE (IDATE FILEDATE)))
        ,@(CL:WHEN SOURCENAME
                (LIST SOURCENAME (GDATE (IDATE SOURCEDATE)))))])
```

(SOURCE-FOR-COMPILED-P

```
  [LAMBDA (SOURCE COMPILED DFASLMARGIN)                   ; Edited  9-May-2022 20:28 by rmk
                                                          ; Edited 25-Apr-2022 08:46 by rmk
```

```
                                                                    ; Edited 31-Oct-2020 09:12 by rmk:
    ;; There seems to be some variation between the source dates in dfasl files and the filecreated date in the sources, they often don't match exactly.
    ;; But if they are within DFASLMARGIN, we assume a match.  We require exact date match for LCOMS

    ;; This is needed for dfasl files created before they recorded the source filecreated name and date instead of the directory source name and date
    ;; when compile took place.

    ;;

    ;; DFASLMARGIN is a pair (after before) where we assume a match if the compiled date is no more than after minutes after the source date and no
    ;; more than before minuts before (the diff is negative then).

    ;; A single positive integer x is interpreted as (x 0).  A single negative integer x is interpreted as (-x x) (before or after x).

    ;; Default is (20 0).

    ;; T is positive or negative infinity
    (CL:UNLESS (LISTP SOURCE)
        (SETQ SOURCE (CREATED-AS SOURCE)))
    (CL:UNLESS (LISTP COMPILED)
        (SETQ COMPILED (CREATED-AS COMPILED)))
    (SETQ DFASLMARGIN (IF (NULL DFASLMARGIN)
                          THEN ;; If compiled is later than source by less than 20 minutes, it's probably OK
                               '(20 0)
                        ELSEIF (EQ T DFASLMARGIN)
                          THEN '(T 0)
                        ELSEIF (LISTP DFASLMARGIN)
                        ELSEIF (NOT (FIXP DFASLMARGIN))
                          THEN (ERROR "ILLEGAL DFASLMARGIN" DFASLMARGIN)
                        ELSEIF (MINUSP DFASLMARGIN)
                          THEN (LIST (MINUS DFASLMARGIN)
                                     DFASLMARGIN)
                        ELSE (LIST DFASLMARGIN 0)))
    (OR (EQUAL (CAR SOURCE)
               (CADDR COMPILED))
        (EQUAL (CADR SOURCE)
               (CADDDR COMPILED))
        (AND (STRING.EQUAL 'DFASL (FILENAMEFIELD.STRING (CAR COMPILED)
                                                        'EXTENSION))
             (LET ((TIMEDIFF (COMPILE-SOURCE-DATE-DIFF COMPILED SOURCE)))
                  ;; If compiled was no more than 20 minutes later, it's probably OK.  Of no more than DFASLMARGIN earlier, if it is negative.
                  (AND (OR (EQ T (CAR DFASLMARGIN))
                           (LEQ TIMEDIFF (CAR DFASLMARGIN)))
                       (OR (EQ T (CADR DFASLMARGIN))
                           (GEQ TIMEDIFF (CADR DFASLMARGIN])
```

(COMPILE-SOURCE-DATE-DIFF

```
  [LAMBDA (CFILE SFILE)                                             ; Edited 20-Sep-2020 22:59 by rmk:
    ;; Positive means that compiled is later than source, normal order but maybe by too much. Negative means that compiled came before source, i.e.,
    ;; compiled on a source that didn't yet exist.

    ;; Value is in minutes
    (ROUND (FQUOTIENT [IDIFFERENCE [IDATE (CADDDR (OR (LISTP CFILE)
                                                      (CREATED-AS CFILE]
                                  (IDATE (CADR (OR (LISTP SFILE)
                                                   (CREATED-AS SFILE]
                   (TIMES 60 ONESECOND])

)

(DEFINEQ
```

(FIX-DIRECTORY-DATES

```
  [LAMBDA (FILES MARGIN)                                            ; Edited 26-Mar-2024 21:42 by rmk
                                                                    ; Edited 29-Nov-2021 20:30 by rmk:
                                                                    ; Edited 23-Nov-2021 12:16 by rmk:
                                                                    ; Edited 30-Oct-2020 22:01 by rmk:
    ;; For Lisp source and compiled files, ensures that the directory file date corresponds to the  filecreated date. Returns the list of files whose dates
    ;; were changed.  For compiled files, it could be that the current directory date was set improperly because of the confusing about the fact that
    ;; FILEDATE (without CFLG) returns the filedate of the source file, not the compiled file itself.  Another fix so that it doesn't do the HELP if it
    ;; discovers that the directory has the source date

    ;;

    ;; Note that (FILEDATE <compiledfile>) returns the filecreated date of the source, not of the compiled file.  (FILEDATE <compiledfile> T) returns the
    ;; date that we actually want.  We could check on the extension, but the safer thing, perhaps, is to ask for first for the compiled date on every file,
    ;; and use if if it isn't NIL.  If it is NIL, then ask for the source date.

    ;;

    ;; Really, there should be a FILEDATE entry that isn't confused in this way, internally figures out the date that the file itself was created

    ;;

    ;; This allows for the fact that directory dates that  are no later than, say, 30 seconds of the filecreated date are probably OK--the directory date
    ;; may be set when the file is closed.

    ;; Use IDATEs in case FDCDATE is not Y2K.
```

```
;; Stop if directory date is more than 2 minutes earlier than the filecreated date.  Earlier could be because the dates are asserted at different points
;; in the filing process.  But 2 minutes is worth thinking about.  Returning from HELP will get them aligned.
;;
;; Doesn't descend into subdirectories.
;;
;; Also fixes dates of Tedit files, if they carry an internal creation date.
(SETQ MARGIN (ITIMES (OR MARGIN 2)
                     60 ONESECOND))
(FOR F DIDATE FCDATE IN (OR (LISTP FILES)
                            (FILDIR FILES))
   UNLESS (DIRECTORYNAMEP F) WHEN (SETQ FCDATE (OR (FILEDATE F T)
                                                   (FILEDATE F)
                                                   (TEDIT.FILEDATE F)))
   UNLESS (IEQP (SETQ DIDATE (GETFILEINFO F 'ICREATIONDATE))
                (SETQ FCDATE (IDATE FCDATE)))
   COLLECT (CL:WHEN (IGREATERP (IDIFFERENCE FCDATE DIDATE)
                               MARGIN)
                ;; If a previous pass use the source date for a compiled file, fix it.  Otherwise, something is odd.
                (CL:UNLESS (IEQP DIDATE (IDATE (FILEDATE F)))
                    (HELP "DIRECTORY DATE EARLIER THAN FILECREATED DATE" (LIST F (GDATE DIDATE)
                                                                               (GDATE FCDATE)))))
         (SETFILEINFO F 'ICREATIONDATE FCDATE)
         F])
```

(FIX-EQUIV-DATES

```
[LAMBDA (CDVALUE)                                                    ; Edited  8-Dec-2021 10:22 by rmk
                                                                     ; Edited 22-Nov-2021 22:31 by rmk:
                                                                     ; Edited  5-Nov-2021 16:49 by rmk:
                                                                     ; Edited  1-Sep-2020 16:21 by rmk:
;; For every entry whose files are EQUIVALENT and whose filedates are different, sets the directory of the file with the later date to be the date of
;; the one with the earlier date.  This preumes that the later one must have been a copy.
(for CDE EARLY LATE in (fetch CDENTRIES of (CD.INSURECDVALUE CDVALUE)) when (fetch EQUIV of CDE)
   unless (EQ '= (fetch DATEREL of CDE)) collect (SELECTQ (fetch DATEREL of CDE)
                                                          (> (SETQ EARLY (fetch INFO2 of CDE))
                                                             (SETQ LATE (fetch INFO1 of CDE)))
                                                          (< (SETQ EARLY (fetch INFO1 of CDE))
                                                             (SETQ LATE (fetch INFO2 of CDE)))
                                                          (SHOULDNT))
                                                 (SETFILEINFO (fetch (CDINFO FULLNAME) OF LATE)
                                                        'ICREATIONDATE
                                                        (GETFILEINFO (fetch (CDINFO FULLNAME) OF EARLY)
                                                               'ICREATIONDATE))
                                                 (fetch (CDINFO FULLNAME) OF LATE])
```

(COPY-COMPARED-FILES

```
[LAMBDA (CDVALUE TARGET MATCHNAMES)                                  ; Edited 22-Nov-2021 22:39 by rmk:
                                                                     ; Edited  5-Nov-2021 16:53 by rmk:
                                                                     ; Edited  1-Sep-2020 16:20 by rmk:
;; Copies source files to target files whose matchname belongs to MATCHNAMES, if given.
;; TARGET is 1 or 2, indicating which side of the CD entry is the target.  Value is the list of matchnames whose files have been copied.
;; Directory filedates and other properties are preserved.
(CL:UNLESS (MEMB TARGET '(1 2))
       (ERROR "INVALID TARGET" TARGET))
(CL:UNLESS CDVALUE
    (PRINTOUT T "Note:  Using LASTCDVALUE" T)
    (SETQ CDVALUE LASTVALUE))
(SETQ MATCHNAMES (MKLIST MATCHNAMES))
(for CDE SINFO TINFO MATCHNAME in (fetch CDENTRIES of CDVALUE) eachtime (SETQ SINFO (fetch INFO1 of CDE))
                                                                        (SETQ TINFO (fetch INFO2 of CDE))
                                                                        (CL:WHEN (EQ TARGET 1)
                                                                               (swap SINFO TINFO))
                                                                        (SETQ MATCHNAME (fetch MATCHNAME
                                                                                               of CDE))
   when (AND (fetch (CDINFO FULLNAME) OF SINFO)
             (fetch (CDINFO FULLNAME) OF TINFO))
   unless (AND MATCHNAMES (NOT (MEMB MATCHNAME MATCHNAMES))) collect (COPYFILE (fetch (CDINFO FULLNAME)
                                                                                       OF SINFO)
                                                                                (PACKFILENAME 'VERSION NIL
                                                                                       'BODY
                                                                                       (fetch (CDINFO FULLNAME)
                                                                                          OF TINFO)))
                                                                               MATCHNAME])
```

(COPY-MISSING-FILES

```
[LAMBDA (CDVALUE TARGET MATCHNAMES)                                  ; Edited 10-Dec-2021 21:56 by rmk
                                                                     ; Edited 22-Nov-2021 22:32 by rmk:
                                                                     ; Edited  5-Nov-2021 16:55 by rmk:
                                                                     ; Edited  1-Sep-2020 16:21 by rmk:
```

```
    ;; Copies source files to target files whose matchname belongs to MATCHNAMES, if given.
    ;; TARGET is 1 or 2, indicating which side of the CD entry is the target.  Value is the list of matchnames whose files have been copied.
    ;; Directory filedates and other properties are preserved.
    (CL:UNLESS (MEMB TARGET '(1 2))
           (ERROR "INVALID TARGET" TARGET))
    (SETQ CDVALUE (CD.INSURECDVALUE CDVALUE))
    (SETQ MATCHNAMES (MKLIST MATCHNAMES))
    (for CDE SINFO TINFO TDIR MATCHNAME (TDIR _ (CL:IF (EQ TARGET 1)
                                                       (fetch (CDVALUE CDDIR1) of CDVALUE)
                                                       (fetch (CDVALUE CDDIR2) of CDVALUE)))
       in (fetch CDENTRIES of CDVALUE) eachtime (SETQ MATCHNAME (fetch MATCHNAME of CDE))
                                          (SETQ SINFO (fetch INFO1 of CDE))
                                          (SETQ TINFO (fetch INFO2 of CDE))
                                          (CL:WHEN (EQ TARGET 1)
                                                   (swap SINFO TINFO))
       when (AND (fetch (CDINFO FULLNAME) OF SINFO)
                 (NOT (fetch (CDINFO FULLNAME) OF TINFO)))
       unless (AND MATCHNAMES (NOT (MEMB MATCHNAME MATCHNAMES))) collect
                                                                    ;; Using the source fullname in the target should
                                                                    ;; preserve the version number
                                                                    (COPYFILE (fetch (CDINFO FULLNAME)
                                                                                 OF SINFO)
                                                                              (PACKFILENAME 'BODY TDIR
                                                                                     'BODY
                                                                                     (fetch (CDINFO FULLNAME)
                                                                                        OF SINFO)))
                                                                    MATCHNAME])
```

(COMPILED-ON-SAME-SOURCE

```
  [LAMBDA (CDVALUE)                                                 ; Edited 22-Nov-2021 22:40 by rmk:
                                                                    ; Edited  5-Nov-2021 16:55 by rmk:
                                                                    ; Edited  9-Sep-2020 13:00 by rmk:
    ;; Returms a subset of CDENTRIES consisting of files that are compiled on the same source (i.e. their source names or dates are the same).
    ;; Preserves the header.
    (CDSUBSET CDVALUE (FUNCTION (LAMBDA (CDE)
                                  (DECLARE (USEDFREE INFO1 INFO2))
                                  (LET (CREATED1 CREATED2)
                                       (CL:WHEN [AND (EQ 'COMPILED (fetch (CDINFO TYPE) OF INFO1))
                                                     (EQ 'COMPILED (fetch (CDINFO TYPE) OF INFO2))
                                                     [CDDR (SETQ CREATED1 (CREATED-AS (fetch (CDINFO FULLNAME)
                                                                                         OF INFO1]
                                                     (CDDR (SETQ CREATED2 (CREATED-AS (fetch (CDINFO FULLNAME)
                                                                                         OF INFO2]
                                                (OR (EQUAL (CADDR CREATED1)
                                                           (CADDR CREATED2))
                                                    (EQUAL (CADDDR CREATED1)
                                                           (CADDDR CREATED2))))])

)

(RPAQ ONESECOND (IDIFFERENCE (IDATE "1-Jan-2020 12:00:01")
                             (IDATE "1-Jan-2020 12:00:00")))

(RPAQ? LASTCDVALUE NIL)

;; Compare-directories browser

(DEFINEQ
```

(CDBROWSER

```
  [LAMBDA (CDVALUE TITLE COLHEADINGS BROWSERPROPS SEPARATEDIRECTIONS MENUITEMS)
    ;; Edited 28-Jan-2022 17:01 by rmk: a table browser for the differences in CDVALUE.
    ;; Creates a table browser for the differences in CDVALUE.
    (SETQ MENUITEMS (IF MENUITEMS
                        THEN (FOR I IN MENUITEMS COLLECT (OR (LISTP I)
                                                             (SASSOC I CDTABLEBROWSER.MENUITEMS)
                                                             (AND (STREQUAL I "")
                                                                  "")
                                                             (ERROR "UNKNOWN CDBROWSER MENU ITEM" I)))
                        ELSE CDTABLEBROWSER.MENUITEMS))
    (LET ((STRINGS (CDBROWSER.STRINGS CDVALUE COLHEADINGS SEPARATEDIRECTIONS))
          WINDOW BROWSER REGION ITEMWIDTH MENUWIDTH)
         (CL:WHEN STRINGS
             ;; Suggest a width that will show all the items
             (SETQ ITEMWIDTH (FOR PAIR IN STRINGS LARGEST (STRINGWIDTH (CAR PAIR)
                                                                       DEFAULTFONT)
                                FINALLY (RETURN $$EXTREME)))
             [SETQ MENUWIDTH (FOR I IN MENUITEMS LARGEST (STRINGWIDTH (CAR (MKLIST I))
                                                                      DEFAULTFONT)
```

```
                        FINALLY (RETURN (WIDTHIFWINDOW (IMAX $$EXTREME (STRINGWIDTH " CD commands "
                                                                              DEFAULTFONT]
```

;; 2 allows for the prompt window

```
[SETQ REGION (GETREGION (PLUS TB.LEFT.MARGIN ITEMWIDTH (TIMES 2 WBorder)
                              MENUWIDTH)
                       (TIMES (IPLUS 2 (IMAX (IMIN 15 (LENGTH STRINGS))
                                             (LENGTH MENUITEMS)))
                              (FONTPROP DEFAULTFONT 'HEIGHT]
(SETQ WINDOW (CREATEW REGION (OR TITLE (CONCAT "Compare directories       " (LENGTH STRINGS)
                                               " files"))
                       NIL T))
(WINDOWPROP WINDOW 'UNDERCONSTRUCTION T)
```

;; TABLEBROWSER is odd: USERDATA is a single recognized property. But it allows for other unrecognized properties in the list, it ;; pushes them on to a list USERPROPS...and then throws it away. So here I'm using USERDATA to hold the directory lengths so they ;; can be stripped off for display. It may actually be better to have a field name in CDVALUE for all of the shared stuff in front of the ;; entries, and keep it all.

```
[SETQ BROWSER (TB.MAKE.BROWSER (FOR PAIR IN STRINGS COLLECT (CD.TABLEITEM PAIR))
                       WINDOW
                       `(PRINTFN CD.TABLEITEM.PRINTFN COPYFN CD.TABLEITEM.COPYFN USERDATA
                              ,(APPEND BROWSERPROPS (LIST 'CDVALUE CDVALUE]
(ATTACHMENU (CREATE MENU
                    TITLE _ " CD commands "
                    MENUFONT _ DEFAULTFONT
                    CENTERFLG _ T
                    ITEMS _ MENUITEMS
                    WHENSELECTEDFN _ (FUNCTION CDTABLEBROWSER.WHENSELECTEDFN))
       WINDOW
       'RIGHT
       'TOP T)
(WINDOWPROP WINDOW 'UNDERCONSTRUCTION NIL)
(GETPROMPTWINDOW WINDOW)
(OPENW WINDOW)
BROWSER)])
```

(CDBROWSER.STRINGS

```
[LAMBDA (CDVALUE COLHEADINGS SEPARATEDIRECTIONS)                    ; Edited 14-Aug-2022 12:13 by rmk
                                                                    ; Edited 11-Aug-2022 20:23 by rmk
                                                                    ; Edited 25-Jul-2022 15:31 by rmk
                                                                    ; Edited 20-Jul-2022 21:14 by rmk
                                                                    ; Edited 22-Feb-2022 18:30 by rmk
                                                                    ; Edited 14-Dec-2021 21:03 by rmk
                                                                    ; Edited  8-Dec-2021 11:22 by rmk
                                                                    ; Edited 27-Nov-2021 21:37 by rmk:
```

;; Create a list of elements one for each CDENTRY of the form (printstring CDENTRY LATER)

;; Wouldn't have to fool around with the stream if there was an option for CDPRINT to return the list of formatted strings.

;; If SEPARATEDIRECTIONS, groups the files that would go from left to right from the files that would go from right to left, with a blank in the ;; middle

```
(CL:UNLESS CDVALUE (SETQ CDVALUE LASTCDVALUE))
(CL:WHEN (FETCH CDENTRIES OF CDVALUE)
    (LET ((SCRATCHSTREAM (OPENSTREAM '{NODIRCORE} 'OUTPUT))
          PREAMBLE COLHEADERS PAIRS L2R R2L BROWSER OBJWINDOW HEADINGW HEADINGHEIGHT)
         (CDPRINT CDVALUE SCRATCHSTREAM COLHEADINGS)
         (OPENSTREAM SCRATCHSTREAM 'INPUT)
         (SETQ PREAMBLE (BIND LINE UNTIL [EQ 0 (NCHARS (SETQ LINE (CL:READ-LINE SCRATCHSTREAM] COLLECT
                                                                                               LINE))
         (CL:WHEN COLHEADINGS
             (SETQ COLHEADERS (CL:READ-LINE SCRATCHSTREAM)))
         (SETQ PAIRS (BIND LATER UNTIL (EOFP SCRATCHSTREAM) AS CDENTRY IN (FETCH CDENTRIES OF CDVALUE)
                        COLLECT (SETQ LATER (SELECTQ (FETCH DATEREL OF CDENTRY)
                                                     (> 'LEFT)
                                                     (< 'RIGHT)
                                                     ((* ?)
                                                          (IF (FETCH INFO1 OF CDENTRY)
                                                              THEN 'LEFT
                                                            ELSE 'RIGHT))
                                                     ((R C)      ; Renamed or copied
                                                          (FETCH DATEREL OF CDENTRY))
                                                     (SHOULDNT)))
                                ;; Take off the EQUIV field.  Should used COL1START
                                (LIST (SUBSTRING (CL:READ-LINE SCRATCHSTREAM)
                                                 2)
                                      CDENTRY LATER)))
         (CL:WHEN SEPARATEDIRECTIONS
             (FOR PAIR IN PAIRS DO (SELECTQ (CADDR PAIR)
                                            (LEFT (PUSH L2R PAIR))
                                            (RIGHT (PUSH R2L PAIR))
                                            (SHOULDNT)))
             (CL:WHEN (AND L2R R2L)
                 ;; Stick a blank object between
                 (SETQ PAIRS (NCONC (DREVERSE L2R)
```

```
                                        [COPY '(("")
                                                ("")]
                                        (DREVERSE R2L)))))
            (CL:WHEN COLHEADERS
                (PUSH PAIRS (LIST COLHEADERS)))
            PAIRS))])

)

;; TABLEBROWSER browser

(FILESLOAD (SYSLOAD)
       TABLEBROWSER)

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)
       TABLEBROWSER)
)

(DEFINEQ
```

(CD.TABLEITEM

```
  [LAMBDA (DATA)                                          ; Edited 27-Nov-2021 22:09 by rmk:
    (CREATE TABLEITEM
           TIDATA _ DATA
           TIUNSELECTABLE _ (NOT (CADR DATA])
```

(CD.TABLEITEM.PRINTFN

```
  [LAMBDA (BROWSER ITEM WINDOW)                           ; Edited 27-Nov-2021 21:38 by rmk:
    (PRIN3 (CAR (FETCH TIDATA OF ITEM))
           WINDOW])
```

(CD.TABLEITEM.COPYFN

```
  [LAMBDA (CDBROWSER ITEM)                                ; Edited 24-Feb-2022 21:12 by rmk
                                                          ; Edited 25-Dec-2021 12:58 by rmk
    (LET [LEFT RIGHT FILE (CDENTRY (CADR (FETCH TIDATA OF ITEM]
         (SETQ LEFT (FETCH (CDINFO FULLNAME) OF (FETCH (CDENTRY INFO1) OF CDENTRY)))
         (SETQ RIGHT (FETCH (CDINFO FULLNAME) OF (FETCH (CDENTRY INFO2) OF CDENTRY)))
         (SETQ FILE (IF (AND LEFT RIGHT)
                        THEN (SELECTQ [MENU (CREATE MENU
                                                    TITLE _ "Which File?"
                                                    ITEMS _ '(Left Right]
                                    (Left LEFT)
                                    (Right RIGHT)
                                    NIL)
                      ELSE (OR LEFT RIGHT)))
         (CL:WHEN FILE
             (PUTCLIPBOARD FILE)
             (COPYINSERT FILE))])
```

(CDTABLEBROWSER.HEADING.REPAINTFN

```
  [LAMBDA (WINDOW REGION)                                 ; Edited 28-Nov-2021 09:09 by rmk:
    (MOVETOUPPERLEFT WINDOW)
    (PRIN3 (WINDOWPROP WINDOW 'COLHEADINGSTRING)
           WINDOW])

)

(DEFINEQ
```

(CDTABLEBROWSER.WHENSELECTEDFN

```
  [LAMBDA (ITEM MENU KEY)                                 ; Edited 28-Nov-2021 20:56 by rmk:
                                                          ; Edited 21-Jan-88 11:40 by bvm
    (ADD.PROCESS `(,(FUNCTION CD.COMMANDSELECTEDFN)
                   ',ITEM
                   ',MENU
                   ',KEY)
           'NAME
           (PACK* 'CD- (CAR ITEM))
           'BEFOREEXIT
           'DON'T])
```

(CD.COMMANDSELECTEDFN

```
  [LAMBDA (MENUITEM MENU KEY)                             ; Edited  6-Mar-2022 19:52 by rmk
                                                          ; Edited 24-Feb-2022 19:52 by rmk
                                                          ; Edited  5-Feb-2022 17:23 by rmk
                                                          ; Edited 27-Jan-2022 17:46 by rmk
                                                          ; Edited 10-Jan-2022 22:51 by rmk
                                                          ; Edited 25-Dec-2021 11:20 by rmk
                                                          ; Edited 12-Jan-87 12:57 by bvm:
    ;; Cobbled from FB.COMMANDSELECTEDFN.  But here we assume that the menu item is of the form (display-string FN . EXTRAS), we peel out
    ;; the FN to apply, leave the rest alone.
```

```
(DECLARE (SPECVARS MENUITEM MENU KEY))
(CL:UNLESS (STREQUAL MENUITEM "")                              ; For blank lines
    (RESETLST
        [LET* [(WINDOW (WINDOWPROP (WFROMMENU MENU)
                           'MAINWINDOW))
               (PWINDOW (GETPROMPTWINDOW WINDOW))
               (CDBROWSER (WINDOWPROP WINDOW 'TABLEBROWSER))
               (USERDATA (TB.USERDATA CDBROWSER))
               (CDVALUE (LISTGET USERDATA 'CDVALUE))
               (FN (CADR (LISTP MENUITEM)))
               (MIDDLE (EQ KEY 'MIDDLE]
              (DECLARE (SPECVARS WINDOW PWINDOW CDVALUE USERDATA))
              (GIVE.TTY.PROCESS PWINDOW)
              (TTYDISPLAYSTREAM PWINDOW)                       ; Pwindow
              (COND
                 ((EQ 0 (TB.NUMBER.OF.ITEMS CDBROWSER 'SELECTED))
                  (FLASHWINDOW PWINDOW)
                  (PRIN3 "Please make a selection" T))
                 (T (CL:WHEN MIDDLE
                        (GIVE.TTY.PROCESS PWINDOW)
                        (CLEARW PWINDOW)
                        (FLASHWINDOW PWINDOW)
                        (CL:UNLESS (EQ 'Y (ASKUSER NIL 'N "Apply to all selected items? "))
                            (SETQ KEY 'LEFT)
                            (PRIN3 " ... " PWINDOW)))
                    (TB.MAP.SELECTED.ITEMS CDBROWSER
                        [FUNCTION (LAMBDA (CDBROWSER TBITEM)
                                    (LET* ((CDENTRY (CADR (FETCH TIDATA OF TBITEM)))
                                           (FILE1 (FETCH (CDINFO FULLNAME)
                                                      (FETCH (CDENTRY INFO1) OF CDENTRY)))
                                           (FILE2 (FETCH (CDINFO FULLNAME)
                                                      (FETCH (CDENTRY INFO2) OF CDENTRY)))
                                           (TYPE (FETCH (CDINFO TYPE) OF (FETCH (CDENTRY INFO1)
                                                                             OF CDENTRY)))
                                           (LABELS (APPLY* (OR (LISTGET USERDATA 'LABELFN)
                                                               (FUNCTION NILL))
                                                           FILE1 FILE2 USERDATA))
                                           (LABEL1 (OR (CAR LABELS)
                                                       FILE1))
                                           (LABEL2 (OR (CADR LABELS)
                                                       FILE2)))
                                          (DECLARE (SPECVARS . T))
                                          ;; One of the files is not real if its date is "", a rename.
                                          (CL:WHEN (STREQUAL "" (FETCH (CDINFO LENGTH)
                                                                    OF (FETCH (CDENTRY INFO1)
                                                                         OF CDENTRY)))
                                                 (SETQ FILE1 NIL))
                                          (CL:WHEN (STREQUAL "" (FETCH (CDINFO LENGTH)
                                                                    OF (FETCH (CDENTRY INFO2)
                                                                         OF CDENTRY)))
                                                 (SETQ FILE2 NIL))
                                          ;; If USERDATA contains a LABELFN, then it is applied to the files and the rest of the USERDATA to
                                          ;; produce abbreviated labels for titles and headers.
                                          (CLEARW T)
                                          (CL:FUNCALL FN TBITEM MENUITEM CDBROWSER KEY]
                        (FUNCTION NILL))
                    (CL:WHEN MIDDLE (PRIN3 " Done" PWINDOW]))])
```

(CD-MENUFN

```
[LAMBDA (TBITEM MENUITEM CDBROWSER KEY)

    ;; Edited 21-May-2022 21:59 by rmk

    ;; Edited 27-Feb-2022 12:47 by rmk: FILELABELFN, if provided, can be used to calculate abbreviated labels for the file, for example, to strip off any
    ;; long common prefixes.

    ;; MENUITEM is of the form (display-atom <this function> . extrainfo).  The selector for the selectq is either the CAR of the extrainfo or the display
    ;; atom.

    (DECLARE (USEDFREE LABEL1 LABEL2 FILE1 FILE2 WINDOW TYPE))
    (SETQ MENUITEM (OR (CADDR MENUITEM)
                       (CAR MENUITEM)))
    (CL:WHEN (MEMB MENUITEM '(Compare See See% right See% both See% left))
                                                               ; Close the previous ones
        (CLOSEWITH.DOIT WINDOW))
    (LET (CHILDREN)
         (SETQ CHILDREN (SELECTQ MENUITEM
                            (Compare (IF (AND FILE1 FILE2)
                                         THEN (CD-COMPARE-FILES FILE1 FILE2 LABEL1 LABEL2 TYPE
                                                    (WINDOWPROP WINDOW 'REGION))
                                       ELSE (FLASHWINDOW T)
                                            (PRIN3 "Only one file" T)))
                            (See% left (IF FILE1
                                           THEN (TEDIT-SEE FILE1 (RELCREATEREGION
                                                                   700 700 'RIGHT 'TOP `(,WINDOW 0.5)
```

```
                                                    (IPLUS (FETCH (REGION BOTTOM)
                                                                  OF (WINDOWPROP WINDOW 'REGION))
                                                           -1)
                                                    T)
                                             NIL
                                             (CONCAT "SEE window for " LABEL1))
                                      ELSE (FLASHWINDOW T)
                                           (PRIN3 "No file to print" T)))
                              (See% right (IF FILE2
                                              THEN (TEDIT-SEE FILE2 (RELCREATEREGION
                                                                     700 700 'LEFT 'TOP `(,WINDOW 0.5)
                                                                     (IPLUS (FETCH (REGION BOTTOM)
                                                                                   OF (WINDOWPROP WINDOW 'REGION))
                                                                            -1)
                                                                     NIL)
                                                    NIL
                                                    (CONCAT "SEE window for " LABEL2))
                                           ELSE (FLASHWINDOW T)
                                                (PRIN3 "No file to print" T)))
                              ((See See% both)
                                   (EXAMINEFILES FILE1 FILE2 LABEL1 LABEL2
                                          (RELCREATEREGION 1400 700 'LEFT 'TOP `(,WINDOW 0.5 -701)
                                                 (IPLUS (FETCH (REGION BOTTOM) OF (WINDOWPROP WINDOW 'REGION))
                                                        -1)
                                                 NIL)))
                              (Copy% -> (CDBROWSER-COPY CDBROWSER TBITEM 'LEFT))
                              (Copy% <- (CDBROWSER-COPY CDBROWSER TBITEM 'RIGHT))
                              (Delete% <- (CDBROWSER-DELETE-FILE CDBROWSER TBITEM 'LEFT T))
                              (|Delete ALL <-|
                                   (CDBROWSER-DELETE-FILE CDBROWSER TBITEM 'LEFT NIL))
                              (Delete% -> (CDBROWSER-DELETE-FILE CDBROWSER TBITEM 'RIGHT T))
                              (|Delete ALL ->|
                                   (CDBROWSER-DELETE-FILE CDBROWSER TBITEM 'RIGHT NIL))
                              (SHOULDNT)))
        (CLOSEWITH CHILDREN WINDOW)
        (MOVEWITH CHILDREN WINDOW])
```

(CD-COMPARE-FILES

```
[LAMBDA (FILE1 FILE2 LABEL1 LABEL2 TYPE PARENTREGION)          ; Edited 22-May-2022 14:41 by rmk
  (PROG NIL
        (SETQ FILE1 (OR (STREAMP FILE1)
                        (INFILEP FILE1)))
        (SETQ FILE2 (OR (STREAMP FILE2)
                        (INFILEP FILE2)))
        (CL:UNLESS TYPE
            (SETQ TYPE (COMPAREDIRECTORIES.INFOS.TYPE FILE1))
            (CL:UNLESS (EQ TYPE (COMPAREDIRECTORIES.INFOS.TYPE FILE2))
                (FLASHWINDOW T)
                (PRIN3 "Can't compare files of different types" T)
                (RETURN)))
        (RETURN (SELECTQ TYPE
                     (SOURCE (CSBROWSER FILE1 FILE2 NIL LABEL1 LABEL2
                                    (AND PARENTREGION (RELCREATEREGION (FIXR (TIMES 0.75 (FETCH (REGION WIDTH)
                                                                                                OF PARENTREGION)))
                                                                       200
                                                                       'LEFT
                                                                       'TOP
                                                                       `(,PARENTREGION 0.125)
                                                                       (IPLUS (FETCH (REGION BOTTOM) OF PARENTREGION)
                                                                              20)
                                                                       NIL))))
                             (COMPILED (FLASHWINDOW T)
                                       (PRIN3 "Cannot compare compiled files" T))
                             ((TEXT TEDIT OTHER)  ;; Works for TEDIT, but doesn't detect image object differences
                                  (LET ((COMPARETEXT.ALLCHUNKS))
                                       (DECLARE (SPECVARS COMPARETEXT.ALLCHUNKS))
                                       (COMPARETEXT FILE1 FILE2 'LINE (AND PARENTREGION
                                                                           (RELCREATEPOSITION
                                                                            `(,PARENTREGION 0.5)
                                                                            (IPLUS (FETCH (REGION BOTTOM) OF
                                                                                                          PARENTREGION
                                                                                          )
                                                                                   20)))
                                                    (LIST LABEL1 LABEL2))))
                             (PROGN (FLASHWINDOW T)
                                    (PRIN3 "Unable to compare, showing both" T)
                                    (EXAMINEFILES FILE1 FILE2 LABEL1 LABEL2 (AND PARENTREGION
                                                                                 (RELCREATEREGION
                                                                                  1400 700 'LEFT 'TOP
                                                                                  `(,PARENTREGION 0.5 -701)
                                                                                  (IPLUS (FETCH (REGION BOTTOM)
                                                                                                OF PARENTREGION)
                                                                                         -1)
                                                                                  NIL])
```

(CDBROWSER-COPY

```
[LAMBDA (CDBROWSER TBITEM SOURCE)                                    ; Edited 24-May-2022 15:49 by rmk
                                                                     ; Edited 25-Apr-2022 09:24 by rmk
                                                                     ; Edited  5-Feb-2022 17:27 by rmk
                                                                     ; Edited  2-Feb-2022 22:18 by rmk
```

;; Copies the file identified as SOURCE (LEFT or RIGHT) in CDENTRY to the other file of the end. If the destination file is missing, it is assumed to ;; be a new/unversioned file of the same name as the source but with the directory prefix switched. CDVALUE needed to know what directory ;; prefixes are involved.

;; Returns NIL if the copy fails.

```
(CL:UNLESS (TB.ITEM.DELETED? CDBROWSER TBITEM)
    (PROG* ((CDVALUE (LISTGET (TB.USERDATA CDBROWSER)
                              'CDVALUE))
            (SOURCEDIR (FETCH (CDVALUE CDDIR1) OF CDVALUE))
            (DESTDIR (FETCH (CDVALUE CDDIR2) OF CDVALUE))
            (CDENTRY (CADR (FETCH TIDATA OF TBITEM)))
            (SOURCEINFO (FETCH (CDENTRY INFO1) OF CDENTRY))
            (DESTINFO (FETCH (CDENTRY INFO2) OF CDENTRY))
            SOURCEFILE DESTFILE SOURCEVER (DATERELBAD '<)
            RESULT)
```

;; Start assuming LEFT, switch if RIGHT

```
           (CL:WHEN (EQ SOURCE 'RIGHT)
               (SWAP SOURCEINFO DESTINFO)
               (SWAP SOURCEDIR DESTDIR)
               (SETQ DATERELBAD '>))
           (SETQ SOURCEFILE (FETCH (CDINFO FULLNAME) OF SOURCEINFO))
           (SETQ DESTFILE (FETCH (CDINFO FULLNAME) OF DESTINFO))
           (CLEARW T)
           (CL:UNLESS SOURCEFILE
               (PRIN3 "No source file to copy" T)
               (RETURN NIL))
           (CL:WHEN [AND (EQ DATERELBAD (FETCH (CDENTRY DATEREL) OF CDENTRY))
                         (PROGN (FLASHWINDOW T)
                                (EQ 'N (ASKUSER NIL NIL "Target is newer than source.  Really copy? "]
                  (RETURN NIL))
           (CL:WHEN [AND (SETQ SOURCEVER (FILENAMEFIELD SOURCE 'VERSION))
                         (ILESSP SOURCEVER (FILENAMEFIELD (INFILEP (PACKFILENAME.STRING 'VERSION NIL
                                                                                       'BODY SOURCEFILE))
                                                          'VERSION))
                         (PROGN (FLASHWINDOW T)
                                (EQ 'N (ASKUSER NIL NIL (CONCAT SOURCEFILE " is not the newest version.
                                                                Really copy? "]
                  (RETURN NIL))
           (CLEARW T)
           (CL:UNLESS DESTFILE
               (SETQ DESTFILE (CD-SWAPDIRS SOURCEFILE SOURCEDIR DESTDIR)))
           (SETQ RESULT (COPYFILE SOURCEFILE (PACKFILENAME.STRING 'VERSION NIL 'BODY DESTFILE)))
           (PRIN3 (IF RESULT
                      THEN (TB.DELETE.ITEM CDBROWSER TBITEM)
                           (CONCAT "Copied to " RESULT)
                    ELSE (FLASHWINDOW T)
                         (CONCAT SOURCEFILE " could not be copied"))
                  T)
           (RETURN RESULT)))])
```

(CDBROWSER-DELETE-FILE

```
[LAMBDA (CDBROWSER TBITEM SIDE ONLYONE SAVE)                         ; Edited 25-Apr-2022 09:06 by rmk
                                                                     ; Edited  5-Feb-2022 17:46 by rmk
                                                                     ; Edited 18-Jan-2022 23:02 by rmk
                                                                     ; Edited 19-Dec-2021 23:33 by rmk
```

;; FILE is a full filename from a CDENTRY, and it will be removed. Unless ONLYONE and FILE has a version number, then all previous versions of ;; the file are also removed so tha the next earliest version doesn't reemerge.

;; The deleted directory should be pruned separately, from time to time.

;; Presumably SAVE is NIL for a git host, since git can restore on its own.

;; If SAVE, then the files are renamed to a deleted directory, not actually expunged, so that they can be restored if needed. The deleted directory is ;; defined by sticking deleted> on the front of FILE's directory.

```
(CL:UNLESS (TB.ITEM.DELETED? CDBROWSER TBITEM)
    [LET ((CDENTRY (CADR (FETCH TIDATA OF TBITEM)))
          FILE OTHERFILE)
         (SETQ FILE (FETCH (CDINFO FULLNAME) OF (FETCH (CDENTRY INFO1) OF CDENTRY)))
         (SETQ OTHERFILE (FETCH (CDINFO FULLNAME) OF (FETCH (CDENTRY INFO2) OF CDENTRY)))
         (CL:WHEN (EQ SIDE 'RIGHT)
                (SWAP FILE OTHERFILE))
         (CL:WHEN FILE
             (FOR F INSIDE (IF (FILENAMEFIELD.STRING FILE 'VERSION)
                               THEN [IF ONLYONE
                                        THEN FILE
                                      ELSE (DREVERSE (FILDIR (PACKFILENAME.STRING 'VERSION "*" 'BODY FILE]
                                    ELSE FILE)
                COLLECT
                        ;; Delete the earlier ones first, if it goes bad, you don't want them to persist. This preserves the original version
```

```
                ;; numbers, maybe it should start fresh from 1 (or from whatever might have been deleted before).
                (IF SAVE
                    THEN (CL:UNLESS (RENAMEFILE F (PACKFILENAME.STRING 'DIRECTORY
                                                                       (CONCAT "deleted>" (FILENAMEFIELD.STRING
                                                                                             F
                                                                                             'DIRECTORY))
                                                                       'BODY F))
                                    (ERROR "Could not delete " F))
                  ELSE (DELFILE FILE))
                F
              FINALLY
                ;; Perhaps only mark it as deleted if both files are gone?
                (TB.DELETE.ITEM CDBROWSER TBITEM)))])])
```

(CD-SWAPDIRS

```
  [LAMBDA (FILE FROMDIR TODIR KEEPVERSION)                    ; Edited 2-Feb-2022 19:10 by rmk
    ;; Replaces prefix FROMDIR of FILE with TODIR
    (IF (STRPOS FROMDIR FILE 1 NIL NIL T FILEDIRCASEARRAY)
        THEN [SETQ FILE (CONCAT TODIR (SUBSTRING FILE (ADD1 (NCHARS FROMDIR]
             (CL:IF KEEPVERSION
                 FILE
                 (PACKFILENAME.STRING 'VERSION NIL 'BODY FILE))
      ELSE (ERROR FILE (CONCAT " doesn't begin with " FROMDIR])

)

(RPAQQ CDTABLEBROWSER.MENUITEMS ((Compare CD-MENUFN)
                                 (Copy% -> CD-MENUFN)
                                 (Copy% <- CD-MENUFN)
                                 (See% left CD-MENUFN)
                                 (See% right CD-MENUFN)
                                 (See% both CD-MENUFN)
                                 (See CD-MENUFN)))

(FILESLOAD (SYSLOAD)
       COMPARESOURCES COMPARETEXT)

(MOVD? 'NILL 'TEDIT.FILEDATE)
```

FUNCTION INDEX

RECORD INDEX

VARIABLE INDEX