File created: 23-May-2022 12:30:29 **{DSK}<users>kaplan>local>medley3.5>working-medley>sources>TESTUPF.;1**

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ TESTUPFCOMS
       ((COMS                                                    ; Original code
              (FNS OLD-UNPACKFILENAME.STRING \UPF.NEXTPOS \UPF.TEMPFILEP)
              (DECLARE%: DONTCOPY (MACROS CANONICAL.DIRECTORY UNPACKFILE1.DIRECTORY UNPACKFILE1)))

        ;; Debugging

        ;; DOTTEDNAMES:  mismatch intended

        ;; RETURNFAILS:  mismatch with DIRFLG=RETURN, DIRECTORY and SUBDIRECTORY are swapped.  But original doesn't agree with its
        ;; own complete analaysis.

        (VARS DOTTEDNAMES TESTS RETURNFAILS)
        (FNS TRY TRYALL DT)))

;; Original code

(DEFINEQ

(OLD-UNPACKFILENAME.STRING
  [LAMBDA (FILE ONEFIELDFLG DIRFLG OSTYPE PACKFLG CLFLG)         ; Edited 25-Jan-2022 17:16 by rmk
                                                                 ; Edited  5-Jan-2022 11:03 by rmk
                                                                 ; Edited 30-Mar-90 22:37 by nm

;;; Given a string or atom representation of a file name, unpack it into its component parts

;;; rmk: devices must come before directories.

    (PROG ((POS 1)
           (LEN (NCHARS FILE))
           TEM BEYONDNAME BEYONDEXT VAL CODE HOSTP SUBDIREND FIRSTDOT SECONDDOT USEDSEMI)
          (COND
             ((NULL FILE)
              (RETURN NIL))
             ((OR (LITATOM FILE)
                  (STRINGP FILE)
                  (NUMBERP FILE)))
             ((TYPEP FILE 'PATHNAME)
              (RETURN (UNPACKPATHNAME.STRING FILE ONEFIELDFLG DIRFLG PACKFLG)))
             [(STREAMP FILE)                                     ; For streams, use full name.  If anonymous, fake it
              (SETQ FILE (OR (ffetch FULLFILENAME of FILE)
                             (RETURN (COND
                                        (ONEFIELDFLG (AND (EQ ONEFIELDFLG 'NAME)
                                                          FILE))
                                        (T (LIST 'NAME FILE]
             (T (\ILLEGAL.ARG FILE)))
          (COND
             ((SELCHARQ (NTHCHARCODE FILE 1)
                   ({                                            ; normal use in Interlisp-D
                      (SETQ TEM (SUB1 (OR (\UPF.NEXTPOS (CHARCODE })
                                                 FILE 2)
                                          0))))
                   (%[                                           ; some Xerox and Arpanet systems use '[' for host
                       (SETQ TEM (SUB1 (OR (\UPF.NEXTPOS (CHARCODE "]")
                                                  FILE 2)
                                           0))))
                   (%(                                           ; this is the 'proposed standard' for Xerox servers
                       (SETQ TEM (SUB1 (OR (\UPF.NEXTPOS (CHARCODE ")")
                                                  FILE 2)
                                           0))))
                   NIL)
              (UNPACKFILE1 'HOST 2 TEM)
              [COND
                 ((EQ TEM -1)                                    ; Started with the host field delimiter, but there was no
                                                                 ; corresponding terminating delimiter .
                                                                 ; I'm not sure why the name is dealt with the host name.

                  (RETURN (DREVERSE VAL]
              (SETQ POS (IPLUS TEM 2))
              [if (EQ OSTYPE T)
                  then                                           ; Use actual host to determine os type
                       (SETQ OSTYPE (GETHOSTINFO (CAR VAL)
                                           'OSTYPE]
              (SETQ HOSTP T)))

     ;; rmk:  if there is a colon before the next < or /, then we must be looking at a device.  A device appears to end after the last colon, i.e., a device
     ;; name can have a colon inside it.

          (COND
             ((AND (SETQ TEM (\UPF.NEXTPOS (CHARCODE (%: < /))
                                    FILE POS))
                   (EQ (CHARCODE %:)
                       (NTHCHARCODE FILE TEM)))                  ; all device returned have DEVICE.END on it so that NIL: will
                                                                 ; work
              (UNPACKFILE1 'DEVICE POS (if CLFLG
                                           then (SUB1 TEM)
                                         else TEM))
              (SETQ POS (ADD1 TEM))
              (SETQ HOSTP T)))
          (COND
             ((EQ DIRFLG 'RETURN)                                ; assert that this is a directory; more forgiving about missing
                                                                 ; trailing delimiter. There are two distinct cases for the missing
                                                                 ; initial delimiter.  If HOST is also specified, it is dealt with as the
                                                                 ; true "relative pathname" by device dependent manner,
                                                                 ; otherwise it is dealt with following the "incomplete file names"
                                                                 ; convention.  In the first case, returns RELATIVEDIRECTORY
                                                                 ; instead of DIRECTORY and in the second case, returns
                                                                 ; SUBDIRECTORY.
              (LET ((TYPE 'DIRECTORY)
                    (START (SELCHARQ (NTHCHARCODE FILE POS)
                                (NIL                             ; just host, return
                                     (RETURN (DREVERSE VAL)))
                                ((/ <)                           ; Started with the initial directory delimiter.
                                     (ADD1 POS))
                                POS))
                    END)
                   (SETQ END (SELCHARQ (NTHCHARCODE FILE -1)
                                  ((/ >)
                                       [COND
                                          ((EQ START POS)        ; Didn't start with a directory delimiter,
                                           (COND
                                              ((NOT HOSTP)       ; "Incomplete file names" case defined in IRM.  This is a
                                                                 ; subdirectory of the current connected directory
                                               (SETQ TYPE 'SUBDIRECTORY))
                                              (T                 ; True "relative pathname".  The way to deal with it is dependent
                                                                 ; on the device on which HOST is implemented.
                                                 (SETQ TYPE 'RELATIVEDIRECTORY]
                                       (COND
                                          ((EQ LEN POS)          ; Only the initial directory is specified (i.e. "{DSK}/").
                                           (SETQ START POS)
                                           -1)
                                          (T -2)))
                                  (PROGN [COND
                                            [(EQ START POS)      ; Both of the initial and trail delimiters are omitted.
                                             (COND
                                                ((NOT HOSTP)     ; "Incomplete file names" case defined in IRM.  This is a
                                                                 ; subdirectory of the current connected directory
                                                 (SETQ TYPE 'SUBDIRECTORY))
                                                (T               ; True "relative pathname".  The way to deal with it is dependent
                                                                 ; on the device on which HOST is implemented.
                                                   (SETQ TYPE 'RELATIVEDIRECTORY]
                                            (T (COND
                                                  ((EQ LEN POS)  ; Only the initial directory is specified (i.e. "{DSK}<").
                                                   (SETQ START POS]
                                         -1)))
                   (UNPACKFILE1.DIRECTORY TYPE START END))
              (RETURN (DREVERSE VAL)))
             ((SELCHARQ (NTHCHARCODE FILE POS)
                   (/                                            ; unix and the 'xerox standard' use / for delimiter
                                                                 ; In the case of the {DSK}/FOO>BAR, FOO should be dealt with
                                                                 ; as a directory.
                      (SETQ TEM (LASTCHPOS (CHARCODE (/ >))
                                       FILE
                                       (ADD1 POS)))
                      T)
                   ((< >)                                        ; Interlisp-D and most other Xerox systems, and Tops-20/Tenex
                                                                 ; use <>.  Jericho uses >>
                                                                 ; In the case of the {DSK}<FOO/BAR, FOO should be dealt with
                                                                 ; as a directory.
                        (SETQ TEM (LASTCHPOS (CHARCODE (> /))
                                         FILE
                                         (ADD1 POS)))
                        T)
                   NIL)

              ;; allow {DSK}/etc   to be a directory specification.

              (if TEM
                  then (UNPACKFILE1.DIRECTORY 'DIRECTORY (ADD1 POS)
                            (SUB1 TEM))
                       (SETQ POS (ADD1 TEM))
                else ;; {DSK}/foo: the directory is /, the name is foo

                    (UNPACKFILE1.DIRECTORY 'DIRECTORY POS POS)
                    (SETQ POS (ADD1 POS)))
              (SETQ HOSTP T))
             ((SETQ TEM (LASTCHPOS (CHARCODE (/ >))
                              FILE POS))                         ;  {eris}abc>  relative

              ;;  This is the true "relative pathname".  Returns RELATIVEDIRECTORY instead of DIRECTORY.
              [COND
                 ((NOT HOSTP)                                             ; "Incomplete file names" case.
                  (UNPACKFILE1.DIRECTORY (if (EQ DIRFLG 'FIELD)
                                              then 'DIRECTORY
                                            else 'SUBDIRECTORY)
                         POS
                         (SUB1 TEM)))
                 (T                                                       ; True "relative pathname".
                    (UNPACKFILE1.DIRECTORY (if (EQ DIRFLG 'FIELD)
                                                then 'DIRECTORY
                                              else 'RELATIVEDIRECTORY)
                           POS
                           (SUB1 TEM]
              (SETQ POS (ADD1 TEM))
              (SETQ HOSTP T)))
          (OR (SETQ CODE (NTHCHARCODE FILE (SETQ TEM POS)))
              (RETURN (DREVERSE VAL)))
          (if (EQ OSTYPE T)
              then                                                        ; There wasn't a host field in the name, so we have no clue
                   (SETQ OSTYPE NIL))
NAMELP

;; At this point, CODE is the TEM'th char of file name.  POS is the first character of the field we are currently working on.
    (SELCHARQ CODE
         (%.                                                        ; Note position for later--we only want to deal with the last set of
                                                                    ; dots
              (if BEYONDNAME
                   then                                             ; no longer of interest (probably a bad name, too)
                 elseif FIRSTDOT
                   then                                             ; We're recording the second dot
                        (if SECONDDOT
                            then                                    ; Note only the two most recent dots
                                 (SETQ FIRSTDOT SECONDDOT))
                        (SETQ SECONDDOT TEM)
                 else (SETQ FIRSTDOT TEM)))
         ((! ; NIL)                                                 ; SUBDIRECTORY, NAME and EXTENSION fields definitely
                                                                    ; terminated by now
               (if (SELCHARQ CODE
                        (!                                          ; ! is only a delimiter on IFS, so ignore it if we know the ostype is
                                                                    ; something else
                           (AND OSTYPE (NEQ OSTYPE 'IFS)))
                        (;                                          ; If we've already parsed the extension, then we have a semi in
                                                                    ; the middle of the version.  Skip it unless it's ;T or ;S
                           [AND BEYONDEXT (NOT (\UPF.TEMPFILEP FILE (ADD1 TEM])
                       NIL)
                  then (GO NEXTCHAR))
               (if FIRSTDOT
                  then                                              ; Have a name and/or extension to parse now
                   (if [AND SECONDDOT
                           (NOT (if OSTYPE
                                    then                            ; Known OS type must be Tops20 for second dot to mean
                                                                    ; version
                                         (EQ OSTYPE 'TOPS20)
                                  else                              ; Unknown OS type, so check that "version" is numeric or
                                                                    ; wildcard
                                      (AND [for I from (ADD1 SECONDDOT) to (SUB1 TEM) bind CH
                                              always (OR (DIGITCHARP (SETQ CH (NTHCHARCODE FILE I)))
                                                         (EQ CH (CHARCODE *]
                                           (SELCHARQ CODE
                                                (NIL                ; end of file name, ok
                                                     T)
                                                (;                  ; This semi-colon better not be introducing a version
                                                   (\UPF.TEMPFILEP FILE (ADD1 TEM)))
                                                NIL]
                      then                                          ; Second dot is not intoducing a version
                           (SETQ FIRSTDOT SECONDDOT)
                           (SETQ SECONDDOT NIL))
                   (UNPACKFILE1 'NAME POS (SUB1 FIRSTDOT))
                   (SETQ POS (ADD1 (if SECONDDOT
                                       then (UNPACKFILE1 'EXTENSION (ADD1 FIRSTDOT)
                                                   (SUB1 SECONDDOT))
                                            (SETQ BEYONDEXT T)
                                            SECONDDOT
                                     else FIRSTDOT)))
                   (SETQ BEYONDNAME T)
                   (SETQ FIRSTDOT NIL))
               (UNPACKFILE1 (COND
                                ((NOT BEYONDNAME)
                                 (SETQQ BEYONDNAME NAME))
                                ((NOT BEYONDEXT)
                                 'EXTENSION)
                                ((AND (EQ BEYONDEXT (CHARCODE ";"))
                                      (\UPF.TEMPFILEP FILE POS)))
                                (T                                  ; Everything after the semi was version
                                   'VERSION))
                     POS
```

```
                           (SUB1 TEM))
                    (if (NULL CODE)
                        then                                     ; End of string
                             (RETURN (DREVERSE VAL)))
                    (SETQ BEYONDEXT CODE)                        ; Note the character that terminated the name/ext
                    (SETQ POS (ADD1 TEM)))
               (%'                                               ; Quoter
                   (add TEM 1))
               NIL)
      NEXTCHAR
          (SETQ CODE (NTHCHARCODE FILE (add TEM 1)))
          (GO NAMELP])

(\UPF.NEXTPOS
  [LAMBDA (CHAR STRING POS)                                      (* lmm " 5-Oct-84 18:41")
    (bind NCH while (SETQ NCH (NTHCHARCODE STRING POS)) do (COND
                                                             ((EQMEMB NCH CHAR)
                                                              (RETURN POS))
                                                             ((EQ NCH (CHARCODE %'))
                                                              (add POS 1)))
                                                          (add POS 1])

(\UPF.TEMPFILEP
  [LAMBDA (FILENAME START)                                       ; Edited  6-Jan-88 13:12 by bvm:

    ;; Checks whether START denotes a temporary mark for Twenex filename beginning at START.  Returns the appropriate field name if so.  Not sure
    ;; we should parse this junk any more, but this at least localizes it.

    (SELCHARQ (NTHCHARCODE FILENAME START)
         ((T S)                                                  ; Funny temp stuff
              (AND (EQ START (NCHARS FILENAME))
                   'TEMPORARY))
         NIL])

)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(PUTPROPS CANONICAL.DIRECTORY MACRO
          [OPENLAMBDA (SRCSTRING)
            (AND
             SRCSTRING
             (LET
              ((LEN (NCHARS SRCSTRING)))
              (COND
                 ((EQ LEN 1)
                  (if (STREQUAL SRCSTRING "/")
                      then "<"
                    else SRCSTRING))
                 (T (LET* ((FATP (ffetch (STRINGP FATSTRINGP) of SRCSTRING))
                           (DSTSTRING (ALLOCSTRING LEN NIL NIL (AND FATP T)))
                           (DSTBASE (ffetch (STRINGP BASE) of DSTSTRING))
                           (DSTPOS 0)
                           (NEXTPOS -1))
                          (if (NOT FATP)
                              then [for SRCPOS from 1 to LEN bind CODE
                                      first (while (EQMEMB (NTHCHARCODE SRCSTRING SRCPOS)
                                                           (CHARCODE (< / >)))
                                               do (add SRCPOS 1))
                                            (if (> SRCPOS LEN)
                                                then (RETURN "<"))
                                      do (SELCHARQ (SETQ CODE (NTHCHARCODE SRCSTRING SRCPOS))
                                              ((> /)
                                                   (if (> DSTPOS NEXTPOS)
                                                       then (\PUTBASETHIN DSTBASE DSTPOS (CHARCODE >))
                                                            (SETQ NEXTPOS (add DSTPOS 1))))
                                              (%' (\PUTBASETHIN DSTBASE DSTPOS CODE)
                                                  (add DSTPOS 1)
                                                  (if (NEQ SRCPOS LEN)
                                                      then (\PUTBASETHIN DSTBASE DSTPOS (NTHCHARCODE SRCSTRING
                                                                                               (add SRCPOS 1)))
                                                           (add DSTPOS 1)))
                                              (PROGN (\PUTBASETHIN DSTBASE DSTPOS CODE)
                                                     (add DSTPOS 1)))
                                      finally (RETURN (if (EQ DSTPOS LEN)
                                                          then (if (EQMEMB (NTHCHARCODE DSTSTRING -1)
                                                                           (CHARCODE (> /)))
                                                                   then (SUBSTRING DSTSTRING 1 -2)
                                                                 else DSTSTRING)
                                                        elseif (EQMEMB (NTHCHARCODE DSTSTRING DSTPOS)
                                                                       (CHARCODE (> /)))
                                                          then (SUBSTRING DSTSTRING 1 (SUB1 DSTPOS))
                                                        else (SUBSTRING DSTSTRING 1 DSTPOS]
                            else (for SRCPOS from 1 to LEN bind CODE
                                   first (while (EQMEMB (NTHCHARCODE SRCSTRING SRCPOS)
                                                      (CHARCODE (< / >)))
                                          do (add SRCPOS 1))
                                   do (SELCHARQ (SETQ CODE (NTHCHARCODE SRCSTRING SRCPOS))
                                          ((> /)
                                               (if (> DSTPOS NEXTPOS)
                                                   then (\PUTBASEFAT DSTBASE DSTPOS (CHARCODE >))
                                                        (SETQ NEXTPOS (add DSTPOS 1))))
                                          (%' (\PUTBASEFAT DSTBASE DSTPOS CODE)
                                              (add DSTPOS 1)
                                              (if (NEQ SRCPOS LEN)
                                                  then (\PUTBASEFAT DSTBASE DSTPOS (NTHCHARCODE SRCSTRING
                                                                                        (add SRCPOS 1)))
                                                       (add DSTPOS 1)))
                                          (PROGN (\PUTBASEFAT DSTBASE DSTPOS CODE)
                                                 (add DSTPOS 1)))
                                   finally (RETURN (if (EQ DSTPOS LEN)
                                                     then (if (EQMEMB (NTHCHARCODE DSTSTRING -1)
                                                                     (CHARCODE (> /)))
                                                             then (SUBSTRING DSTSTRING 1 -2)
                                                           else DSTSTRING)
                                                   elseif (EQMEMB (NTHCHARCODE DSTSTRING DSTPOS)
                                                                 (CHARCODE (> /)))
                                                     then (SUBSTRING DSTSTRING 1 (SUB1 DSTPOS))
                                                   else (SUBSTRING DSTSTRING 1 DSTPOS])

(PUTPROPS UNPACKFILE1.DIRECTORY MACRO [OPENLAMBDA (NAM ST END)
                                          (LET* ((OLDDIR (SUBSTRING FILE ST END))
                                                 (NEWDIR (CANONICAL.DIRECTORY OLDDIR)))
                                                (COND
                                                   [(NOT ONEFIELDFLG)
                                                    (SETQ VAL (CONS (COND
                                                                       (PACKFLG (AND NEWDIR (MKATOM NEWDIR)))
                                                                       (T (OR NEWDIR "")))
                                                                    (CONS NAM VAL]
                                                   ((EQMEMB NAM ONEFIELDFLG)
                                                    (RETURN (COND
                                                               (PACKFLG (AND NEWDIR (MKATOM NEWDIR)))
                                                               (T (OR NEWDIR ""])

(PUTPROPS UNPACKFILE1 MACRO [OPENLAMBDA (NAM ST END)                 (* lmm "22-APR-81 22:21")
                                (COND
                                   [(NOT ONEFIELDFLG)
                                    (SETQ VAL (CONS (COND
                                                       (PACKFLG (SUBATOM FILE ST END))
                                                       (T (OR (SUBSTRING FILE ST END)
                                                              ""))))
                                                    (CONS NAM VAL]
                                   ((EQMEMB NAM ONEFIELDFLG)
                                    (RETURN (COND
                                               (PACKFLG (SUBATOM FILE ST END))
                                               (T (OR (SUBSTRING FILE ST END)
                                                      ""])
)
)
```

;; Debugging

;; DOTTEDNAMES: mismatch intended

;; RETURNFAILS: mismatch with DIRFLG=RETURN, DIRECTORY and SUBDIRECTORY are swapped. But original doesn't agree with its own
;; complete analaysis.

```
(RPAQQ DOTTEDNAMES (".x" ">.git" "x.y.100"))

(RPAQQ TESTS
       ("*,;" "*.*;*" "*.;" "*.;*" "///abc/x" "/abc.x" "<" "<<<abc" "<<<abc>" "<<<abc>>" "<<<abc>x" "<<abc"
              "<<xyz>>>zz" "<<xyz>>>zzz/" "<<xyz>>zz" "<<xyz>zz" "<ABC>" "<XYZ>aa" "<a.b>" "<a;b>" "<ab;c"
              "<ab>" "<abc" "<abc*." "<abc.x" "<abc.x;1" "<abc;x" "<abc<<<x" "<abc<xyz<foo" "<abc<xyz>qrs"
              "<abc>" "<abc>;1" "<abc>xyz" "<abc>xyz>foo" "<xxx" "<xy>>zz" "<xyz>>>zzz/" ">" ">>>abc/x" ">abc"
              ">abc;1" ">abc>" ">abc>xyz>foo" ">xxx" "A.B.C" "XXX<yyy" "a;b" "a;b/d" "a;b;c" "a;b;c;d" "aa"
              "aa;" "aa;NEWEST" "aa;newest" "aaa" "aaa/bbb" "aaa/bbb/" "aaa/xyz;x;m" "aaa<bbb" "aaa<bbb/"
              "aaa<xyz>" "aaa>bbb>" "aaa>xyz.e;m;n" "aaa>xyz>qrs" "abc" "abc...c" "abc///XYZ//" "abc/d"
              "abc/xyz" "abc/xyz.qrs" "abc/xyz.qrs;2" "abc:x<qrs>z" "abc<<<XYZ//" "abc<x" "abc<xyz"
              "abc<xyz>qq" "abc<xyzqq" "abc>;1" "abc>qr.x" "abc>xy" "abc>xyz" "abc>xyz;2" "dev:aaa>xyz>qrs"
              "foo:" "foo:aaa<xyz" "foo:aaa<xyz>" "foo:x<qrs>z" "foo<a:B>" "s;n;b" "x.y.z;w" "x.y;z" "x;y"
              "x<abc<xyz>qrs" "x<abc<z" "x<abc>z" "xxx<yyy" "xxx<yyy>" "xxx<yyy>zzz" "xxx>yyy" "xxx>yyy>"
              "{ABC}" "{ABC}XXX:" "{DSK}" "{DSK}*.;*" "{DSK}...<a" "{DSK}<a" "{DSK}xxx<a" "{DSK}xxx<xxx>yyy"
              "{DSK}xxx>xxx" "{DSK}xxx>yyy" "{HOST}foo:x<qrs>z" "{HOST}x<qrs>z" "{abc}"
              "{dsk}foo:aaa>b>.c.e.g;f" "{dsk}foo:aaa>b>.c.e;f" "{dsk}foo:aaa>b>c.e;f" "{eris}abc>"
              "{host}abc/xyz;2" "{host}abc>xyz;2" "{x}abc<xyz>qq" "{x}abc<xyzqq" "<abc<xyz>abc" "<abc<xyz>qrs"
              "<abc<xyz>"))

(RPAQQ RETURNFAILS (">" ">>>abc/x" ">abc" ">abc;1" ">abc>" ">abc>xyz>foo" ">xxx" ">" ">>>abc/x" ">abc"
                         ">abc;1" ">abc>" ">abc>xyz>foo" ">xxx"))

(DEFINEQ
(TRY
  [LAMBDA (FILE ONEFIELDFLG DIRFLG)                            ; Edited 23-May-2022 12:09 by rmk
                                                               ; Edited 25-Apr-2022 14:15 by rmk
                                                               ; Edited 24-Apr-2022 08:45 by rmk
                                                               ; Edited 21-Apr-2022 15:36 by rmk
    (CL:WHEN (LISTP (CAR (LISTP FILE)))
        (SETQ FILE (CAR FILE)))
    (LET (ORIG NEW)
         (CL:WHEN (LISTP FILE)
             (SETQ ONEFIELDFLG (CADR FILE))
             (SETQ DIRFLG (CADDR FILE))
             (SETQ FILE (CAR FILE)))
         (SETQ ORIG (OLD-UNPACKFILENAME.STRING FILE ONEFIELDFLG DIRFLG))
         (SETQ NEW (UNPACKFILENAME.STRING FILE ONEFIELDFLG DIRFLG))
         (LIST (LIST FILE ONEFIELDFLG DIRFLG)
               (AND (EQUAL ORIG NEW)
                    '=)
               ORIG NEW])

(TRYALL
  [LAMBDA (FILES ALLFLAG ONEFIELDFLG DIRFLG)                   ; Edited 21-Apr-2022 17:56 by rmk
                                                               ; Edited  2-Apr-2022 23:50 by rmk
                                                               ; Edited 31-Mar-2022 22:57 by rmk
    (CL:WHEN (LISTP FILES)
        (SETQ FILES (FOR F IN FILES COLLECT (CL:IF (LISTP (CAR (LISTP F)))
                                                   (CAR F)
                                                   F))))
    (FOR FILE INFO (SAME _ 0)
          (DIFF _ 0) IN FILES EACHTIME (SETQ INFO (TRY FILE ONEFIELDFLG DIRFLG))
                                       (CL:IF (CADR INFO)
                                           (ADD SAME 1)
                                           (ADD DIFF 1))
       UNLESS (AND (CADR INFO)
                   (NOT ALLFLAG))
       COLLECT (PRINTOUT T .P2 (CAAR INFO)
                      31)
             (IF (CADR INFO)
                 THEN (PRINTOUT T "  = " .P2 (CADDR INFO))
                      (CL:WHEN (OR (CADAR INFO)
                                   (CADDAR INFO))
                          (PRINTOUT T 60 (CADAR INFO)
                                 %,,
                                 (CADDAR INFO))
                          (TERPRI T))
               ELSE (PRINTOUT T " ~= " -2 "old: " .P2 (CADDR INFO))
                    (CL:WHEN (OR (CADAR INFO)
                                 (CADDAR INFO))
                        (PRINTOUT T 60 (CADAR INFO)
                               %,,
                               (CADDAR INFO))
                        (TERPRI T))
                    (PRINTOUT T 37 "new: " .P2 (CADDDR INFO)
                           T))
             INFO
       FINALLY (PRINTOUT T SAME " matches, " DIFF " mismatches" T])

(DT
  [LAMBDA (STRINGS ALLFLAG)                                    ; Edited 21-Apr-2022 17:53 by rmk
                                                               ; Edited 19-Apr-2022 20:55 by rmk

    ;; Tests the DIRFLG options on STRINGS.  If an element of STRINGS is a list, it is assumed to be a (STRING ONEFIELD DIRFLG), STRING is
    ;; extracted.

    (SETQ STRINGS (FOR S INSIDE STRINGS COLLECT (CL:IF (LISTP S)
                                                       (CAR S)
                                                       S)))
    [AND NIL (FOR ONEFIELD IN '(NAME DIRECTORY SUBDIRECTORY RELATIVEDIRECTORY)
                JOIN (FOR DIR ORIG NEW SAME IN '(FIELD RETURN)
                        JOIN (PRINTOUT T T "ONEFIELDFLG = " ONEFIELD -3 "DIRFLG = " DIR T T)
                             (TRYALL STRINGS ALLFLAG ONEFIELD DIR))
                FINALLY (FOR INFO SAME (DIFF _ 0) IN $$VAL DO (CL:IF (CADR INFO)
                                                                  (ADD SAME 1)
                                                                  (ADD DIFF 1))
                           FINALLY (SETQ SAME (IDIFFERENCE (LENGTH STRINGS)
                                                           DIFF))
                                   (PRINTOUT T T "Overall:  " SAME " matched, " DIFF " mismatched" T]
    (TRYALL (FOR S IN STRINGS JOIN (FOR ONEFIELD IN '(NAME DIRECTORY SUBDIRECTORY RELATIVEDIRECTORY)
                                      JOIN (FOR DIR IN '(FIELD RETURN) COLLECT (LIST S ONEFIELD DIR])
)
```

-- Listed on 9-Oct-2024 02:49:27 --

FUNCTION INDEX

VARIABLE INDEX

MACRO INDEX