

File created: 26-Jun-90 19:35:02 {DSK}<usr>local>lde>lispcore>internal>library>WHEREIS.;2

changes to: (VARS WHEREISCOMS)

previous date: 29-Oct-86 13:17:33 {DSK}<usr>local>lde>lispcore>internal>library>WHEREIS.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1983, 1984, 1986, 1990 by Venue & Xerox Corporation. All rights reserved.

```
(RPAQQ WHEREISCOMS
(
  (FILES HASH) ; WHEREIS from a hashfile
  (FNS HASHFILE-WHEREIS CLOSEWHEREIS WHEREISNOTICE WHEREISNOTICE1)
  (ADDVARS (WHEREIS.HASH))
  (GLOBALVARS WHEREIS.HASH)
  (DECLARE\ : DONTVAL@LOAD DOCOPY (P (MOVD 'HASHFILE-WHEREIS 'WHEREIS))
    (ADDVARS (AROUNDXITFNS CLOSEWHEREIS)))
  (COMS ; Possibly obsolete now that directory enumerates highest
    ; version ok
    (FNS \REMOVEOLDVERSIONS))
  (LOCALVARS . T)))

;; WHEREIS from a hashfile

(FILESLOAD HASH)

(DEFINEQ

(HASHFILE-WHEREIS
  (LAMBDA (NAME TYPE FILES FN) (* |bvm:| "28-Apr-86 12:24")
    (PROG (VAL) (* |if| FN |given,| APPLY* |to| |each| |element| |and| |return| NIL)
      (COND (* T |as| \a NAME |has| \a |special| |meaning| |to| INFILECOMS?
        ((EQ NAME T)
          |so| |don't| |pass| |through.|)
          (RETURN NIL)))
        (SETQ TYPE (GETFILEPKGTYPE TYPE))
        (|for| FILE |in| (OR (LISTP FILES)
          FILELST)
          |do| (COND
            ((INFILECOMS? NAME TYPE (FILECOMS FILE))
              (COND
                (FN (APPLY* FN NAME FILE)))
                (SETQ VAL (CONS FILE VAL))))))
            (AND (EQ FILES T)
              (EQ TYPE 'FNS)
              (LITATOM NAME)
              (PROGN (COND
                ((AND WHEREIS.HASH (NLISTP WHEREIS.HASH))
                  (* |make| |sure| WHEREIS.HASH |is| \a |list.|)
                  (SETQ WHEREIS.HASH (LIST WHEREIS.HASH))))
                (|for| WHISHSFILE HNAME HSFILE DELP |on| WHEREIS.HASH
                  |do|
                    (* WHEREIS.HASH |is| \a |list| |of| |hash| |file| |names| |off| |of| |which| |the| |hash| |file| |structure| |is| |linked| |into| |the|
                      |system| |hash| |array.| |The| |full| |file| |name| |is| |hashed.|)

                    (COND
                      ((LISTP (SETQ HNAME (CAR WHISHSFILE)))
                        (* |file| |already| |has| |an| |associated| |hashfile| |datatype|)
                        (SETQ HSFILE (CDR HNAME)))
                      ((SETQ HSFILE (FINDFILE HNAME T))
                        (COND
                          ((|find| X |in| WHEREIS.HASH |suchthat| (AND (LISTP X)
                            (EQ HSFILE (HASHFILEPROP (CDR X)
                              'NAME))))
                            (* |Looks| |like| \a |duplicate| |entry|)
                            (RPLACA WHISHSFILE (SETQ HSFILE NIL))
                            (SETQ DELP T))
                          (T (SETQ HSFILE (OPENHASHFILE HSFILE))
                            (* |if| |the| |data| |file| |is| |ever| |closed,| |break| |the| |link| |to| |the| |hash| |file| |structure.|)

                            (WHENCLOSE (HASHFILEPROP HSFILE 'STREAM)
                              'BEFORE
                                (FUNCTION (LAMBDA (STRM)
                                  (|for| TAIL |on| WHEREIS.HASH
                                    |when| (AND (LISTP (CAR TAIL))
                                      (EQ STRM (HASHFILEPROP (CDAR TAIL)
                                        'STREAM))))
                                    |do|
                                      (* |remove| |the| |hashfile| |structure| |for| |this| |file's| |entry| |on|
                                        WHEREIS.HASH.)
```

```

(RPLACA TAIL (CAAR TAIL))))))
'CLOSEALL
'NO)
(RPLACA WHISHSFILE (CONS HNAME HSFILE))))))
(T (OR (EQ 'Y (ASKUSER 120 'Y (CONCAT HNAME " a file on WHEREIS.HASH, not found
-- do you want to delete and continue?")
'((Y "es")
(N "o"))))
(ERRORX (LIST 23 HNAME)))
(RPLACA WHISHSFILE (SETQ HSFILE NIL))
(SETQ DELP T)))
(COND
(HSFILE (|for| FILE |inside| (GETHASHFILE NAME HSFILE)
|when| (NOT (FMEMB FILE VAL))
|do| (AND FN (APPLY* FN NAME FILE))
(|push| VAL FILE))))))
|finally| (COND
(DELP (SETQ WHEREIS.HASH (DREMOVE NIL WHEREIS.HASH))))))
(RETURN (AND (NULL FN)
(DREVERSE VAL))))))

```

(CLOSEWHEREIS

```

(LAMBDA (FLG) (* |bvm:| "28-Apr-86 12:33")

(* * |Close| |the| |whereis| |file| |over| |logout.| |since| |there's| |no| |point| |in| |paying| |to| |keep| |it| |open|)

(AND WHEREIS.HASH (SELECTQ FLG
((NIL BEFORELOGOUT BEFORESYSOUT BEFOREMAKESYS)
(|for| HF |in| (|for| WH |in| WHEREIS.HASH |when| (LISTP WH) |collect|
(* |Gather| |the| |hashfile| |handles|) (CDR WH))
|do| (NLSETQ (CLOSEHASHFILE HF))))
NIL)))

```

(WHEREISNOTICE

```

(LAMBDA (FILEGROUPS NEWFLG DATABASEFILE SCRATCHDIR COMPUTE.HIGHEST.VERSIONS.MANUALLY)
(* |bvm:| "5-Oct-86 16:36")

;; Copies the current whereis hash-file into a scratch file, then notices the files in FILEGROUP The copy is so that this function will execute even
;; though someone else is reading the current database. The database is copied to a scratch file, then renamed to be a newer version of the
;; previous database, which is deleted. This allows others to use the old database while the copying is going on. If an earlier version of the scratch
;; file exists, it means that someone else is currently updating (their version disappears when they complete successfully or logout), so we wait for
;; them to finish.
;; COMPUTE.HIGHEST.VERSIONS.MANUALLY means don't trust DIRECTORY to get the highest version of a file only when enumerating.

(RESETLST
(PROG ((DATABASEFILENAME (OR DATABASEFILE (|if| WHEREIS.HASH
|then| ; if there is a list of files, use the top one.
(|if| (NLISTP WHEREIS.HASH)
|then| WHEREIS.HASH
|elseif| (NLISTP (CAR WHEREIS.HASH))
|then| (CAR WHEREIS.HASH)
|else| (CAAR WHEREIS.HASH))
|else| 'WHEREIS.HASH)))
(SCRATCHVAL (LIST NIL))
HF SCRATCH OLDWH)
(SETQ OLDWH (INFILEP DATABASEFILENAME)) ; creates a scratch file
(|if| (AND OLDWH (NOT NEWFLG))
|then| ; copy old one
(RESETSAVE NIL (LIST (FUNCTION (LAMBDA (X)
(|if| (CAR X)
|then| (CLOSEF? (CAR X))
(AND RESETSTATE (DELFILE (CAR X))))))
SCRATCHVAL))
(RPLACA SCRATCHVAL (SETQ HF (CLOSEF (OPENFILE (SETQ SCRATCH (PACKFILENAME
'DIRECTORY
(FILENAMEFIELD
DATABASEFILENAME
'DIRECTORY)
'NAME
'NEWWHEREISDATABASE
'EXTENSION
'SCRATCH
'TEMPORARY
'S))
'OUTPUT
'NEW))))
; Compensate for the fact that PACKFILENAME produces
; version -1 for temporary ;S

(AND (EQ (SYSTEMTYPE)
'TOPS20)
(SETQ SCRATCH (PACKFILENAME 'VERSION NIL 'BODY SCRATCH))))

;; If there is a version earlier than the one we got, someone else must have it, and we must wait until he gets rid of it (by
;; deleting it)
(|bind| OLDV (RPT _ 1) |until| (EQ HF (SETQ OLDV (FULLNAME SCRATCH 'OLDEST))))

```

```

      |do| (DISMISS 2000)
          (OR (NULL RPT)
              (|if| (EQ RPT 5)
                  |then| (|printout| T T (GETFILEINFO OLDV 'AUTHOR)
                          " seems to be updating the database right now." T "I'm
                          waiting for him to finish." T T)
                      (SETQ RPT NIL)
                      |else| (|add| RPT 1))))
          (SETQ HF (COPYHASHFILE OLDWH HF NIL NIL T))
          (CLOSE? OLDWH)
    |elseif| (AND OLDWH (EQ NEWFLG 'NOCOPY))
    |then| (SETQ HF (OPENHASHFILE OLDWH 'BOTH NIL))
          (SETQ SCRATCHDIR NIL)
    |else| (RESETSAVE NIL (LIST (FUNCTION (LAMBDA (X)
                                      (|if| (CAR X)
                                          |then| (SETQ X (CLOSEHASHFILE (CAR X)))
                                          (AND RESETSTATE (DELFILE X))))
                                SCRATCHVAL))
          (RPLACA SCRATCHVAL (SETQ HF (CREATEHASHFILE (|if| SCRATCHDIR
                                                          |then| (PACKFILENAME.STRING 'DIRECTORY
                                                                    SCRATCHDIR 'BODY
                                                                    DATABASEFILENAME)
                                                          |else| DATABASEFILENAME)
                                                                    'SMALLEXPB NIL (OR (NUMBERP NEWFLG)
                                                                    20000))))
          (SETQ NEWFLG T))

;; Must leave the new file open--otherwise, the user might lose access to it before he starts to do the noticing.
(|for| X |in| (|for| FILESPEC |inside| FILEGROUPS |bind| TEM
              |join| (|if| (SETQ TEM (INFILEP FILESPEC))
                      |then| ; an individual file
                          (LIST TEM)
                      |else| ; a specification for a group of files, expand it. Default to *.; -- i.e., highest version only of the
                          ; extensionless files on this dir.
                          (SETQ TEM (DIRECTORY (PACKFILENAME.STRING 'BODY FILESPEC 'NAME "*"
                                                                    'EXTENSION "" 'VERSION "")))
                          (|if| COMPUTE.HIGHEST.VERSIONS.MANUALLY
                              |then| (\\REMOVEOLDVERSIONS TEM)
                              |else| TEM)))
              |do| (ERSETQ (|printout| T (WHEREISNOTICE1 X HF)
                          -2)))
  (SETQ HF (CLOSEHASHFILE HF))
  (RPLACA SCRATCHVAL NIL)

;; This closes the file, but other updaters are still locked out cause they go for a new version and then trip over our old one.
(COND
  ((NOT NEWFLG)
   (|if| (SETQ HF (RENAMEFILE HF (PACKFILENAME 'VERSION NIL 'BODY DATABASEFILENAME)))
     |then| (DELFILE OLDWH)))
  (SCRATCHDIR (SETQ HF (RENAMEFILE HF DATABASEFILENAME))))
; Now others can get in to read or update.
(RETURN HF))))

```

(WHEREISNOTICE1

```

  (LAMBDA (FILE HF TRYHARD)
    (RESETLST
     (PROG (NAME MAP DATE VAL ENV STREAM)
       (RESETSAVE NIL (LIST 'CLOSE (SETQ STREAM (OPENSTREAM FILE 'INPUT 'OLD NIL ' (
                                                                 DON\ 'T.CHANGE.READ.DATE
                                                                 DON\ 'T.CHANGE.DATE))))
       ))
      (SETQ FILE (FULLNAME STREAM))
      (SETFILEPTR STREAM 0)
      (CL:MULTIPLE-VALUE-SETQ (ENV MAP)
        (GET-ENVIRONMENT-AND-FILEMAP STREAM T))
      (OR MAP (RETURN (LIST FILE "--can't find filemap")))
      (|if| (AND (NOT TRYHARD)
                (EQUAL (SETQ DATE (FILEDATE STREAM))
                      (GETHASHFILE FILE HF))
            DATE)
        |then| ; already analyzed
              (RETURN (LIST FILE DATE)))
      (SETQ NAME (NAMEFIELD FILE T))
      (|for| X |in| (CDR MAP) |do| (|for| Y |in| (CDDR X)
                                          |unless| (OR (NULL (SETQ VAL (LOOKUPHASHFILE (CAR Y)
                                                                 NAME HF ' (INSERT RETRIEVE))))
                                                    (EQ NAME VAL)
                                                    (AND (LISTP VAL)
                                                         (FMEMB NAME VAL))))
                                          |do|
        ;; the first LOOKUPHASHFILE stores NAME as value if there was no previous value, else returns
        ;; previous value. If that value was non-null and did not contain NAME, now have to store union of
        ;; NAME and what was there.
        (PUTHASHFILE (CAR Y)

```

```

(NCONC1 (OR (LISTP VAL)
            (LIST VAL))
        NAME)
(HF)))
(PTHASHFILE FILE DATE HF)
(RETURN FILE))))

```

```
)
```

```
(ADDTovar WHEREIS.HASH )
```

```
(DECLARE\ : DOEVAL@COMPILE DONTCOPY
```

```
(GLOBALVARS WHEREIS.HASH)
```

```
)
```

```
(DECLARE\ : DONTVAL@LOAD DOCOPY
```

```
(MOVD 'HASHFILE-WHEREIS 'WHEREIS)
```

```
(ADDTovar AROUNDEXITFNS CLOSEWHEREIS)
```

```
)
```

:: Possibly obsolete now that directory enumerates highest version ok

```
(DEFINEQ
```

```
(\REMOVEOLDVERSIONS
```

```
(LAMBDA (FULLFILEST)
```

```
(* |rrb| "22-Feb-84 18:12")
```

```
(* |removes| |all| |but| |the| |newest| |version| |of| |any| |file| |on| FULLFILEST.
|Slow| |version| |as| |temporary| |until| DIRECTORY |has| \a |way| |of| |asking| |for| |only| |the| |most| |recent| |version|.|)
```

```
(PROG ((EXPANDEDFILEST ((for FILE in FULLFILEST collect (UNPACKFILENAME FILE)))
```

```
UNIQUELST FILE)
```

```
(for EXPTAIL on EXPANDEDFILEST
```

```
do
```

```
(* |skip| |deleted| |files|.|)
```

```
(AND (SETQ FILE (CAR EXPTAIL))
```

```
(PROG ((XDIRECTORY (LISTGET FILE 'DIRECTORY))
```

```
(XNAME (LISTGET FILE 'NAME))
```

```
(XEXTENSION (LISTGET FILE 'EXTENSION))
```

```
(XVERSION (LISTGET FILE 'VERSION)))
```

```
(* |go| |thru| |the| |list| |of| |expanded| |files| |and| |see| |if| |there| |are| |any| |other| |files| |on| |the| |list| |with| |the| |same|
|name|. |If| |so| |and| |it| |is| |older,| |delete| |it|. |If| |so| |and| |it| |is| |newer,| |don't| |copy| |this| |guy| |onto| |the| |result| |list|.|)
```

```
((for EFTAIL on (CDR EXPTAIL) do (SETQ FILE (CAR EFTAIL))
```

```
(COND
```

```
((AND (EQ (LISTGET FILE 'NAME)
          XNAME)
```

```
(EQ (LISTGET FILE 'EXTENSION)
```

```
XEXTENSION)
```

```
(EQ (LISTGET FILE 'DIRECTORY)
```

```
XDIRECTORY))
```

```
(COND
```

```
((IGREATERP (LISTGET FILE 'VERSION)
             XVERSION)
```

```
(* XFILE |should| |be| |deleted|)
```

```
(RETURN NIL))
```

```
(T
```

```
(* |mark| |it| |deleted|.| |Don't| |want| |to| |play| |around| |with| |the| |pointers| |because| |the| |enclosing| FOR |is| |using| |the|
|same| |list|.|)
```

```
(RPLACA EFTAIL NIL))))))
```

```
|finally| (SETQ UNIQUELST (CONS (CAR EXPTAIL)
```

```
UNIQUELST))))))
```

```
(RETURN ((for UFILE in UNIQUELST collect (PACKFILENAME UFILE))))))
```

```
)
```

```
(DECLARE\ : DOEVAL@COMPILE DONTCOPY
```

```
(LOCALVARS . T)
```

```
)
```

```
(PUTPROPS WHEREIS COPYRIGHT ("Venue & Xerox Corporation" 1983 1984 1986 1990))
```

FUNCTION INDEX

CLOSEWHEREIS	2	WHEREISNOTICE	2	\\REMOVEOLDVERSIONS	4
HASHFILE-WHEREIS	1	WHEREISNOTICE1	3		

VARIABLE INDEX

AROUNDEXITFNS	4	WHEREIS.HASH	4
---------------------	---	--------------------	---
