

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\ : DONTEVAL@LOAD DOCOPY (P (MOVD 'HASHFILE-WHEREIS 'WHEREIS))
    (ADDVARS (AROUNDEXITFNS 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
      ((EQ NAME T) (* T |as| \a NAME |has| \a |special| |meaning| |to| INFILECOMS?)
        |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 HSFIL 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 HSFIL (CDR HNAME)))
                    ((SETQ HSFIL (FINDFILE HNAME T))
                      (COND
                        ((|find| X |in| WHEREIS.HASH |suchthat| (AND (LISTP X)
                          (EQ HSFIL (HASHFILEPROP (CDR X)
                            'NAME))))
                          (* |Looks| |like| \a |duplicate| |entry|)
                          (RPLACA WHISHSFILE (SETQ HSFIL NIL))
                          (SETQ DELP T))
                        (T (SETQ HSFIL (OPENHASHFILE HSFIL))))
                      (COND
                        ((WHENCLOSE (HASHFILEPROP HSFIL '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))))

```



(NCONC1 (OR (LISTP VAL)
(LIST VAL))
NAME)
HF)))

(PUTHASHFILE FILE DATE HF)
(RETURN FILE))))

)

(ADDTOVAR WHEREIS.HASH )

(DECLARE\ : DOEVAL@COMPILE DONTCOPY

(GLOBALVARS WHEREIS.HASH)

)

(DECLARE\ : DONTEVAL@LOAD DOCOPY

(MOVD 'HASHFILE-WHEREIS 'WHEREIS)

(ADDTOVAR AROUNDEXITFNS CLOSEWHEREIS)

)

:: Possibly obsolete now that directory enumerates highest version ok

(DEFINEQ

(\REMOVEOLDVERSIONS

(LAMBDA (FULLFILELST

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

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

(PROG ((EXPANDEDFILELST (|for| FILE |in| FULLFILELST |collect| (UNPACKFILENAME FILE)))
UNIQUELST FILE)

(|for| EXPTAIL |on| EXPANDEDFILELST

|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| EFLTAIL |on| (CDR EXPTAIL) |do| (SETQ FILE (CAR EFLTAIL))

(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 EFLTAIL 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
---------------------	---	--------------------	---

---