

File created: 3-Apr-86 21:33:56 {LOGOS:AFB:SIP}<DOUG>LISP>ACCESS.;3

changes to: (VARS ACCESSCOMS)
(FNS ACCESS)

previous date: 1-Apr-86 17:15:26 {LOGOS:AFB:SIP}<DOUG>LISP>ACCESS.;2

Read Table: OLD-INTERLISP-FILE

Package: INTERLISP

Format: XCCS

(* * Copyright (c) 1986 by Speech Input Project, Univ. of Edinburgh.
All rights reserved.)

```
(RPAQQ ACCESSCOMS ((FNS ACCESS ACCESS.PARSE ACCESS.PARSE.DIRECTORY ACCESS.DIRECTORYNAME ACCESS.MKDIR
ACCESS.PARSE.OBJ ACCESS.PARSE.ACCESS ACCESS.DO.COMMAND ACCESS.OPEN ACCESS.SHOW
ACCESS.ADD ACCESS.REMOVE ACCESS.CHANGE ACCESS.SLASHIFY.DIRNAME ACCESS.NUM.TO.STRING
ACCESS.STRING.TO.NUM)))
```

(DEFINEQ

(ACCESS

[LAMBDA NIL

(* drc: " 3-Apr-86 21:32")

(* Top level function. Parses a command line, checks args and performs commands.)

```
(bind (USER/PWD _ (\INTERNAL/GETPASSWORD NIL NIL NIL NIL NIL 'NS))
COM DIR OBJ ACCESS VALUE do [SETQ VALUE (NLSETQ (LET ((COMMAND (ACCESS.PARSE DIR OBJ ACCESS USER/PWD)
))
(SELECTQ (SETQ COM (CAR COMMAND))
(ABORT (PRINTOUT T "[aborted]"))
(Q (TERPRI T))
(L (SETQ USER/PWD (CDR COMMAND)))
(P (TERPRI T)
(USEREEXEC '___))
(PROGN (for X in (CDR COMMAND) as Y
in ' (DIR OBJ ACCESS)
do (SET Y X))
(ACCESS.DO.COMMAND USER/PWD COM DIR
OBJ ACCESS]
[if (NLISTP VALUE)
then (LET ((ERROR (ERRORN)))
(* printout error messages, but treat CONROL-E errors as
aborts)
(if (EQ (CAR ERROR)
47)
then (PRINTOUT T "[aborted]")
else (ERRORMESS ERROR)
until (EQ COM 'Q])
```

(ACCESS.PARSE

[LAMBDA (DIR.DEFAULT OBJ.DEFAULT ACCESS.DEFAULT USER/PWD)

(* drc: " 1-Apr-86 17:09")

(* Prompts for and parses one command line. Simulates ASKUSER.)

```
(RESETFORM (CONTROL T)
(PROG (CHAR COMMAND DIR OBJ ACCESS USER PWD)
PARSE
(FRESHLINE T)
(printout T "FS: ")
(SETQ CHAR (CHCON1 (READC T T)))
(SELCHARQ CHAR
((L l)
(printout T "ogin")
(SETQQ COMMAND L)
(GO LOGIN))
((S s)
(printout T "how")
(SETQQ COMMAND S))
((A a)
(printout T "dd")
(SETQQ COMMAND A))
((R r)
(printout T "emove")
(SETQQ COMMAND R))
((C c)
(printout T "hange")
(SETQQ COMMAND C))
((P p)
(printout T "ush (type OK to return)")
(RETURN (LIST 'P)))
(^Y)
(printout T " (type OK to return)")
(RETURN (LIST 'P)))
```

```

((Q q)
  (printout T "uit")
  (RETURN (LIST 'Q)))
((H h ?)
  (printout T " one of:" T " L - Login," T " S - Show," T " A - Add," T " R -
    Remove," T " C - Change," T " P, ^Y - Push," T "or Q - Quit.")
  (GO PARSE))
((CR LF)
  (GO PARSE))
(PROGN (PRIN1 (CHARACTER (CHARCODE ^G)))
  (GO PARSE))
(SETQ DIR (ACCESS.PARSE.DIRECTORY DIR.DEFAULT USER/PWD))
(OR DIR (RETURN ' (ABORT)))
[SELECTQ COMMAND
  (S
    (RETURN (LIST COMMAND DIR)))
  (PROGN (SETQ OBJ (ACCESS.PARSE.OBJ OBJ.DEFAULT COMMAND))
    (OR OBJ (RETURN ' (ABORT)))
    (SELECTQ COMMAND
      (R
        (RETURN (LIST COMMAND DIR OBJ)))
      (PROGN (SETQ ACCESS (ACCESS.PARSE.ACCESS ACCESS.DEFAULT COMMAND))
        (if ACCESS
          then (RETURN (LIST COMMAND DIR OBJ ACCESS))
          else (RETURN ' (ABORT)))
        (* done w/ add parse)
      )
    )
  ]
  (* done w/ Remove parse)
  (* done w/ List parse)
)

LOGIN
[SETQ USER (MKATOM (PROMPTFORWORD " (username)" (CAR USER/PWD)
  NIL NIL NIL NIL (LIST (CHARCODE CR)
    (CHARCODE LF]
  (OR USER (RETURN ' (ABORT)))
  (SETQ PWD (\ENCRYPT.PWD (PROMPTFORWORD " (password)" NIL NIL NIL "**")))
  (OR PWD (RETURN ' (ABORT)))
  (RETURN (CONS COMMAND (CONS USER PWD))

```

(ACCESS.PARSE.DIRECTORY

```

[LAMBDA (DIR.DEFAULT USER/PWD DON'T.CHECK)
  (* drc: " 1-Apr-86 16:13")
  (DECLARE (GLOBALVARS \CONNECTED.DIRECTORY))
  (LET ((DIR (PROMPTFORWORD " (access to directory)" DIR.DEFAULT "the name of an NS directory")))
    (if (NOT DIR)
      then
        (* user just typed CR)
        NIL
      else
        (* default host to connected host)
        (OR (FILENAMEFIELD DIR 'HOST)
          (SETQ DIR (MKSTRING (PACKFILENAME 'HOST (FILENAMEFIELD \CONNECTED.DIRECTORY 'HOST)
            'DIRECTORY DIR]
          )
        (if DON'T.CHECK
          then DIR
          elseif (ACCESS.DIRECTORYNAME DIR USER/PWD)
          else (printout T " not an NS directory.")
            NIL])

```

(ACCESS.DIRECTORYNAME

```

[LAMBDA (HOST/DIR USER/PWD)
  (* drc: " 1-Apr-86 16:35")
  (if (AND (STRPOS ":" HOST/DIR)
    (DIRECTORYNAMEP HOST/DIR))
    then HOST/DIR
    elseif [LET ((POS (STRPOS ">" HOST/DIR)))
      (* there are two >'s in HOST/DIR --
        could be a non-existent subdirectory)
      (AND POS (STRPOS ">" HOST/DIR (ADD1 POS)
        then (SELECTQ (ASKUSER DWIMWAIT 'Y (CONCAT " Create subdirectory " HOST/DIR " ? "))
          (Y (ACCESS.MKDIR HOST/DIR USER/PWD))
          NIL])

```

(ACCESS.MKDIR

```

[LAMBDA (HOST/DIR USER/PWD)
  (* drc: " 1-Apr-86 16:54")
  (RESETLST
    (LET* ((HOST (FILENAMEFIELD HOST/DIR 'HOST))
      (DIR (ACCESS.SLASHIFY.DIRNAME (FILENAMEFIELD HOST/DIR 'DIRECTORY)))
      [PARENT (CONCATLIST (DREVERSE (CDR (FMEMB '/' (DREVERSE (UNPACK DIR)
        (CONNECTION (ACCESS.OPEN HOST PARENT USER/PWD))
        (STREAM (CAR CONNECTION))
        (SESSION (CADR CONNECTION))
        (HANDLE (CADDR CONNECTION))
        (COURIER.CALL STREAM 'FILING 'CREATE HANDLE (BQUOTE ((NAME , DIR)
          (IS.DIRECTORY T)
          (FILE.TYPE 1)))
          NIL SESSION 'RETURNERRORS)
        HOST/DIR))])

```

(ACCESS.PARSE.OBJ

```

[LAMBDA (OBJ.DEFAULT COMMAND)
  (* drc: "28-Mar-86 13:24")

```



```
" access)"]])
```

(ACCESS.ADD

```
[LAMBDA (NSNAME HOST/DIR ACCESS OLD.LIST STREAM HANDLE SESSION)
(* drc: "15-Mar-86 13:13")
  (LET ((TRIPLE (if (CH.RETRIEVE.ITEM NSNAME 'USER)
                    then (LIST NSNAME 'INDIVIDUAL ACCESS)
                    elseif (CH.RETRIEVE.ITEM NSNAME 'USERGROUP)
                    then (LIST NSNAME 'GROUP ACCESS)
                    else (ERROR NSNAME "NOT AN NS OBJECT"))))
    VALUE)
  (SETQ VALUE (COURIER.CALL STREAM 'FILING 'CHANGE.ATTRIBUTES HANDLE
    (LIST (LIST 'ACCESS.LIST (LIST (CONS TRIPLE OLD.LIST)
    NIL)))
    SESSION 'RETURNERRORS))
  (if (EQ (CAR VALUE)
    'ERROR)
    then (printout T T VALUE " Not added.")
    else (printout T T "OK, " NSNAME " added to " HOST/DIR " with " (ACCESS.NUM.TO.STRING ACCESS)
    " access."]))
```

(ACCESS.REMOVE

```
[LAMBDA (NSNAME HOST/DIR OLD.LIST STREAM HANDLE SESSION)
(* drc: "15-Mar-86 13:14")
  (LET ([ENTRY (bind (NAME.STRING _ (NSNAME.TO.STRING NSNAME)) for X in OLD.LIST
    thereis (STREQUAL NAME.STRING (NSNAME.TO.STRING (CAR X)
    VALUE)
    (if ENTRY
      then (SETQ VALUE (COURIER.CALL STREAM 'FILING 'CHANGE.ATTRIBUTES HANDLE
        (LIST (LIST 'ACCESS.LIST (LIST (REMOVE ENTRY OLD.LIST)
        NIL)))
        SESSION 'RETURNERRORS))
      (if (EQ (CAR VALUE)
        'ERROR)
        then (printout T T VALUE " Not removed.")
        else (printout T T "OK, " NSNAME " removed from access list of " HOST/DIR "."))
    else (printout T T NSNAME " not on access list for " HOST/DIR ".")])
```

(ACCESS.CHANGE

```
[LAMBDA (NSNAME HOST/DIR ACCESS OLD.LIST STREAM HANDLE SESSION)
(* drc: "13-Mar-86 10:49")
  (LET ([OLD.ENTRY (bind (NAME.STRING _ (NSNAME.TO.STRING NSNAME)) for X in OLD.LIST
    thereis (STREQUAL NAME.STRING (NSNAME.TO.STRING (CAR X)
    NEW.ENTRY VALUE)
    (if OLD.ENTRY
      then (SETQ NEW.ENTRY (LIST NSNAME (CADR OLD.ENTRY)
        ACCESS))
      (SETQ VALUE (COURIER.CALL STREAM 'FILING 'CHANGE.ATTRIBUTES HANDLE
        (LIST (LIST 'ACCESS.LIST (LIST (SUBST NEW.ENTRY OLD.ENTRY OLD.LIST)
        NIL)))
        SESSION 'RETURNERRORS))
      (if (EQ (CAR VALUE)
        'ERROR)
        then (printout T T VALUE " Access not changed.")
        else (printout T T "OK, " NSNAME "'s access to " HOST/DIR " changed to " (
        ACCESS.NUM.TO.STRING ACCESS)
        "."))
    else (printout T T NSNAME " not on access list for " HOST/DIR ".")])
```

(ACCESS.SLASHIFY.DIRNAME

```
[LAMBDA (DIR)
(* drc: "1-Apr-86 15:33")
  (CONCATLIST (SUBST '/' '>' (UNPACK DIR))
```

(ACCESS.NUM.TO.STRING

```
[LAMBDA (NUM)
(* edited: "11-Mar-86 11:24")
  (* * Converts a numeric access code to a string representation)
  (CONCATLIST (for MASK in ' (16 8 4 2 1) as PROTECTION in ' (R W C A D) when (BITTEST NUM MASK) collect
    PROTECTION
  ])
```

(ACCESS.STRING.TO.NUM

```
[LAMBDA (STRING)
(* edited: "11-Mar-86 11:16")
  (* * Converts from a string to a numeric representation of an access code.
  Does not do much error checking. Error is signalled by a returning 0)
  (APPLY (FUNCTION IPLUS)
    (for CHAR in (UNPACK (U-CASE STRING)) collect (SELECTQ CHAR
    (D 1)
```

```
{MEDLEY}<lispusers>ACCESS.;1  (ACCESS.STRING.TO.NUM cont.)
```

Page 5

```
(A 2)  
(C 4)  
(W 8)  
(R 16)  
(RETURN NIL))
```

```
)  
(PUTPROPS ACCESS COPYRIGHT ("Speech Input Project, Univ. of Edinburgh" 1986))
```

FUNCTION INDEX

ACCESS	1	ACCESS.DO.COMMAND	3	ACCESS.PARSE	1	ACCESS.REMOVE	4
ACCESS.ADD	4	ACCESS.MKDIR	2	ACCESS.PARSE.ACCESS	3	ACCESS.SHOW	3
ACCESS.CHANGE	4	ACCESS.NUM.TO.STRING	4	ACCESS.PARSE.DIRECTORY ..	2	ACCESS.SLASHIFY.DIRNAME .	4
ACCESS.DIRECTORYNAME	2	ACCESS.OPEN	3	ACCESS.PARSE.OBJ	2	ACCESS.STRING.TO.NUM	4
