

File created: 27-Sep-90 11:03:58 {FUJI1:MV:ENVOS}<LISPARS>ADMIN>SMART-TRICKLE.;5

changes to: (FNS |Smart-Trickle| FILE-SERVER-UP-P)

previous date: 25-Sep-90 18:14:07 {FUJI1:MV:ENVOS}<LISPARS>ADMIN>SMART-TRICKLE.;1

Read Table: XCL

Package: INTERLISP

Format: XCCS

; Copyright (c) 1990 by Venue. All rights reserved.

```
(RPAQQ SMART-TRICKLECOMS ((FILES (SYSLOAD)
                             PROMPTREMINDERS COPYFILES)
                          (FNS FILE-SERVER-UP-P |Smart-Trickle| |TrickleProcessLogfile|)
                          (ADVISE |Trickle|)
                          (DECLARE\ : DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR (ADDVARS (NLAMA)
                                                                                             (NLAML)
                                                                                             (LAMA)
                                                                                             FILE-SERVER-UP-P
                                                                                             )))))
```

```
(FILESLOAD (SYSLOAD)
            PROMPTREMINDERS COPYFILES)
```

```
(DEFINEQ
```

```
(FILE-SERVER-UP-P
```

```
(CL:LAMBDA (SERVER &OPTIONAL (MESSAGE NIL MESSAGEP)) ; Edited 27-Sep-90 10:50 by gadener
```

```
;; Checks for a certain period of time if server is up. It will try to find a directory specific for the type of server.
; Edited 27-Sep-90 10:45 by gadener
```

```
(CL:BLOCK FILE-SERVER-UP-P
  (CL:UNLESS (NULL SERVER)
    (LET* ((DIRECTORYNAME (COND
      ((CL:SEARCH ":" SERVER) ; This is an NS-server
      "DESKTOPS")
      ((CL:SEARCH "/" SERVER) ; This is an NFS-server
      "/" )
      ((AND (EQL (MACHINETYPE)
                  'MAIKO)
            (OR (STRING-EQUAL SERVER "DSK")
                (STRING-EQUAL SERVER "UNIX")))) ; This should be local disk
      "/" )
      (T ; Assume it is an IFS-server
       "SYSTEM"))))
      (PROCESS-RESULT (CONS))
      (PROCESS-HANDLE (ADD.PROCESS ` (COND
        ((DIRECTORYNAMEP ,DIRECTORYNAME ,SERVER)
         (RPLACA ' ,PROCESS-RESULT T)))
        'NAME "file-server-upp"))))
    (DISMISS 500)
    (|forDuration| 30 |timerUnits| 'SECONDS |untill| (CAR PROCESS-RESULT) |do| (DISMISS 500)
      |finally| (DEL.PROCESS PROCESS-HANDLE))
    (CL:IF (CAR PROCESS-RESULT)
      T
      (CL:WHEN MESSAGEP
        (CL:FORMAT T ">>~%>> ~A but server ~A is down ~%>>~%" MESSAGE SERVER)
        (MENU (CREATE MENU
          TITLE _ (CL:CONCATENATE 'STRING " Server " (STRING SERVER)
                                   " is down ")
          ITEMS _ ' ((WAIT T "Wait until server is up")
                     (CONDITIONS:CONTINUE NIL "Continue to load the rest"))
          CENTERFLG _ T))))))
```

```
(|Smart-Trickle|
```

```
(LAMBDA (|Source| |Destination| |RootLogfileName| |MailAddress| |ScheduleAnotherOne| |DontReplaceOldVersions|
        |DontCopyExtensions| |Retries|) ; Edited 27-Sep-90 11:03 by gadener
```

;; The Smart-Trickle will first check to see that the file-server of both the Source and Destination are up and running before trying to copy files between
;; them . If either is down, it will retry every 15 minutes up to four times before giving up. If it fails to connect and ScheduleAnotherOne has a value
;; other than () , then it will reschedule for that time, 24 hrs later.

```
(LET ((RETRY-TIMES 4) ; How many retries.
      (RETRY-PERIOD 15) ; How many minutes between retries
      |DateString| |LogfileName|)
  (IF (NOT (AND (OR |Retries| (SETQ |Retries| RETRY-TIMES))
```

```
; This is the first time around
```

```
(FILE-SERVER-UP-P (STRING (FILENAMEFIELD |Source| 'HOST)))
```

```
(FILE-SERVER-UP-P (STRING (FILENAMEFIELD |Destination| 'HOST))))
```

```
THEN
```

```
;; At least one of the servers is not up, reschedule a new Smart-trickle to try again in 15 minutes , more correctly , the value of  
;; RETRY-PERIOD
```

```

(IF (> |Retries| 0)
  THEN
    ;; Just Retry
    (SETREMINDER (MKATOM (CONCAT "Trickle-Retry-" (GENSYM)
                                "-" |Source|))
      NIL
      `(|Smart-Trickle| , |Source| , |Destination| , |RootLogfileName| , |MailAddress|
        , |ScheduleAnotherOne|
        , |DontReplaceOldVersions|
        , |DontCopyExtensions|
        , (- |Retries| 1))
      (GDATE (PLUS (IDATE)
                    (TIMES 60 RETRY-PERIOD))))
    ELSE
      ;; Could not connect to either, or both of the servers after retrying several times.
      (PRINTOUT T "Trickle: Tried to connect to " HOST1 " and " HOST2 " , but one or both,
        are down!" T (IF |ScheduleAnotherOne|
          THEN "Rescheduling another one for tomorrow!")))
ELSE
  ;; OK, servers were up, run the Trickler.
  (SETQ |DateString| (DATE (DATEFORMAT SPACES NUMBER.OF.MONTH NO.TIME)))
  (SETQ |LogfileName| (PACK* (OR |RootLogfileName| ' {|UNIX}<tmp>Trickle-log-|)
    (SUBSTRING |DateString| 7 8)
    (SUBSTRING |DateString| 4 5)
    (SUBSTRING |DateString| 1 2)
    '.COPYLOG))
  (IF (EQ '\ (NTHCHAR |LogfileName| -10))
    THEN (SETQ |LogfileName| (MKATOM (RPLSTRING |LogfileName| -10 "0"))))
  ;; ensure that the logfile has one line per file operated on
  (RESETVAR FILELINELENGTH 1000 (COPYFILES |Source| |Destination|
    (APPEND (LIST '>A (LIST 'OUTPUT |LogfileName|)
      (|if| (NULL |DontReplaceOldVersions|)
        then (LIST 'REPLACE))
      (|if| |DontCopyExtensions|
        then `((DONTCOPY ,@|DontCopyExtensions|))))))
    (|TrickleProcessLogfile| |LogfileName| |MailAddress| |Source| |Destination|))
  (IF |ScheduleAnotherOne|
    THEN
      ;; No matter what happened earlier, if a new Trickle has to be scheduled, DO IT.
      (SETREMINDER (MKATOM (CONCAT "Trickle-" (GENSYM)
                                "-" |Source|))
        NIL
        `(|Smart-Trickle| , |Source| , |Destination| , |RootLogfileName| , |MailAddress|
          , |ScheduleAnotherOne| , |DontReplaceOldVersions| , |DontCopyExtensions|)
        (CONCAT (SUBSTRING (GDATE (PLUS (IDATE)
                                          (TIMES 60 60 24)))
            1 10)
          (|if| (EQ |ScheduleAnotherOne| T)
            then (CONCAT (RAND 1 5)
              ":"
              (RAND 0 59))
            else |ScheduleAnotherOne|))))))

(|TrickleProcessLogfile|
  (LAMBDA (|LogfileName| |MailAddress| |Source| |Destination|) (* |N.H.Briggs| " 1-Oct-87 11:29")
    (PROG (|LogfileStream| |EndsOfLines| (|EOLCharacter| (CHARACTER (CHARCODE EOL)))
      |EndOfLine| |Deletions|)
      (SETQ |LogfileStream| (OPENTEXTSTREAM |LogfileName|))
      (|if| (ZEROP (GETEOFPTR |LogfileStream|))
        then
          (* * |probably| |errored| -
            |people| |don't| |usually| |Trickle| |empty| |directories|)
          (|if| |MailAddress|
            then (TEDIT.INSERT |LogfileStream| (PACK* "Subject: (Error?) Trickle:" |Source| " to "
              |Destination| |EOLCharacter| "To: "
              |MailAddress| |EOLCharacter| |EOLCharacter|)
              1)
              (LAFITE.SENDMESSAGE |LogfileStream|)
              (CLOSEF |LogfileStream|)
              (RETURN))
            (TEDIT.SETSEL |LogfileStream| 1 1 'LEFT)
            (SETQ |EndsOfLines| (CONS 0 (|while| (SETQ |EndOfLine| (TEDIT.FIND |LogfileStream| |EOLCharacter|))
              |collect| (TEDIT.SETSEL |LogfileStream| |EndOfLine| 1 'RIGHT)
              |EndOfLine|)))
          (* * |find| |lines| |with| "skipped" |and| |collect| |for| |deletion|)
          (* TEDIT.FIND |is| |very| |poor| |on| |long| |files.| |see| AR# 4220)
          (|for| |EndOfPreviousLine| |on| |EndsOfLines| |bind| |StartOfLine| |EndOfLine|
            |eachtime| (SETQ |StartOfLine| (AND |EndOfPreviousLine| (ADD1 (CAR |EndOfPreviousLine|))))
            (SETQ |EndOfLine| (CADR |EndOfPreviousLine|)))

```

```

|when| (AND |EndOfLine| (STRPOS "skipped" (TEDIT.SEL.AS.STRING |LogfileStream|
(TEDIT.SETSEL |LogfileStream| |StartOfLine|
(ADD1 (IDIFFERENCE |EndOfLine| |StartOfLine|
))))))

```

```

|do|

```

```

(* * |if| |this| |deletion| |is| |an| |extension| |of| |the| |previous| |one|,| |then| |extend| |the| |previous| |one|,| |otherwise| |add| |this|
|to| |the| |collection|. |This| |collapsing| |makes| |the| |actual| |deletion| |much| |more| |efficient|,| |since| |we| |expect| |to| |have|
|few| |of| |the| |lines| |kept|. )

```

```

(|if| (AND |Deletions| (EQUAL (PLUS (CAAR |Deletions|)
(CDAR |Deletions|))
|StartOfLine|))
|then| (RPLACD (CAR |Deletions|)
(PLUS (CDAR |Deletions|)
(DIFFERENCE |EndOfLine| (CAR |EndOfPreviousLine|))))
|else| (|push| |Deletions| (CONS |StartOfLine| (DIFFERENCE |EndOfLine| (CAR |EndOfPreviousLine|)
))))))

```

```

(* * |do| |collected| |deletions|)

```

```

(|for| |Deletion| |in| |Deletions| |do| (TEDIT.DELETE |LogfileStream| (CAR |Deletion|)
(CDR |Deletion|)))

```

```

(* * KLUDGE! TEDIT.PUT |bombs| |after| |putting| |the| |file| |if| |the| |stream| |doesn't| |have| \a |window| |associated|)

```

```

(NLSETQ (TEDIT.PUT |LogfileStream| |LogfileName| T T))

```

```

(* * |construct| \a |mail| |message| |and| |send| |it|)

```

```

(|if| |MailAddress|
|then| (TEDIT.INSERT |LogfileStream| (PACK* (|if| (NOT (ZEROP (GETEOFPTR |LogfileStream|)))
|then| "Subject: Trickle: "
|else| "Subject: (Empty) Trickle:")
|Source| " to " |Destination| |EOLCharacter| "To: "
|MailAddress| |EOLCharacter| |EOLCharacter|)
1)
(LAFITE.SENDMESSAGE |LogfileStream|))
(CLOSEF |LogfileStream|)))

```

```

)

```

```

(XCL:REINSTALL-ADVICE ' |Trickle| :AROUND ' ((:LAST (|Smart-Trickle| |Source| |Destination| |RootLogfileName|
|MailAddress| |ScheduleAnotherOne|
|DontReplaceOldVersions| |DontCopyExtensions|))))

```

```

(READWISE |Trickle|)

```

```

(DECLARE\ : DONTVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR

```

```

(ADDTOTVAR NLAMA )

```

```

(ADDTOTVAR NLAML )

```

```

(ADDTOTVAR LAMA FILE-SERVER-UP-P)

```

```

)

```

```

(PUTPROPS SMART-TRICKLE COPYRIGHT ("Venue" 1990))

```

FUNCTION INDEX

FILE-SERVER-UP-P	1	Smart-Trickle 	1	TrickleProcessLogfile 	2
------------------------	---	----------------------	---	------------------------------	---

ADVICE INDEX

Trickle 	3
----------------	---
