```
(RPAQQ ARADMINCOMS
       (
;; Functions for helping do the routine chores of AR Administration.  Set up to run things automatically.
        (VARS (IDLE.PROFILE '(TIMEOUT NIL FORGET NIL ALLOWED.LOGINS NIL DISPLAYFN (LAMBDA (W)
                                                                              (IDLE.BOUNCING.BOX
                                                                               W "Auto AR Cleanup
                                                                               Machine")))))
              (IDLE.TIMEOUT NIL)
              (IDLE.DEFAULTMODE 'UNLOCKED))
        (ADDVARS (INITIALSLST (SYBALSKY |John| |jds|)))
        (VARS (LOGINHOST/DIR '|{AR:MV:ENVOS}<LISPARS>ADMIN|)
              (SYSOUTGAG T)
              (PROMPT#FLG T)
              (DEFAULTREGISTRY '|envos|)
              (CH.DEFAULT.DOMAIN 'MV)
              (CH.DEFAULT.ORGANIZATION 'ENVOS)
              (DIRECTORIES '("{DSK}/USR/LOCAL/LDE/LISPCORE/SOURCES/" "{DSK}/USR/LOCAL/LDE/LISPCORE/LIBRARY/"
                             "{DSK}/USR/LOCAL/LDE/LISPCORE/INTERNAL/LIBRARY/"
                             "{DSK}/USR/LOCAL/LDE/LISPUSERS/" "{DSK}<usr>local>lde>rooms>sources>"
                             "{DSK}<usr>local>lde>rooms>users-src>" {DSK}))
              (DISPLAYFONTDIRECTORIES '("{DSK}/usr/local/lde/fonts/display/publishing/"
                                        "{DSK}/usr/local/lde/fonts/display/presentation/"
                                        "{DSK}/usr/local/lde/lispcore/XeroxPrivate/Fonts"
                                        "{DSK}/usr/local/lde/fonts/display/chinese/"
                                        "{DSK}/usr/local/lde/fonts/display/JIS1/"
                                        "{DSK}/usr/local/lde/fonts/display/JIS2/"
                                        "{DSK}/usr/local/lde/fonts/display/miscellaneous/"
                                        "{DSK}/usr/local/lde/fonts/display/printwheel/"))
              (DISPLAYFONTEXTENSIONS '(AC DISPLAYFONT STRIKE))
              (INTERPRESSFONTDIRECTORIES '("{DSK}/usr/local/lde/fonts/interpress/publishing/"
                                           "{DSK}/usr/local/lde/fonts/interpress/presentation/"
                                           "{DSK}/usr/local/lde/lispcore/XeroxPrivate/Fonts"
                                           "{DSK}/usr/local/lde/fonts/interpress/chinese/"
                                           "{DSK}/usr/local/lde/fonts/interpress/JIS1/"
                                           "{DSK}/usr/local/lde/fonts/interpress/JIS2/"
                                           "{DSK}/usr/local/lde/fonts/interpress/miscellaneous/"
                                           "{DSK}/usr/local/lde/fonts/interpress/printwheel/"))
              (DEFAULTPRINTINGHOST '("Twister:" "Tremor:")))
;;
;; Tools to load
;;
        (DECLARE\: DONTEVAL@LOAD DOCOPY (FILES PROMPTREMINDERS CROCK WHO-LINE FILEWATCH)
               (P (LAFITEMODE 'NS)
                  (LOGOW "AR & Trickle")
                  (CROCK (CREATEREGION 900 800 100 100))))
;;
;;  The AR stuff
;;
;; Every other day AR.CLEANUP to create new summaries &c
;;
        (FNS PERIODIC-AR-CLEANUP DO-PERIODIC-AR-CLEANUP)
;; Weekly creation of the list of ARs to review.
        (FNS PERIODIC-AR-REVIEW DO-WEEKLY-AR-REVIEW)
        (FNS MY-FLOOR)
        (DECLARE\: DONTEVAL@LOAD DOCOPY (FILES ARCLEANUP ARHACK)
               (VARS (STARTING-MONDAY (IDATE "5-DEC-88 07:00:00"))
                     (ONE-WEEK 604800)
                     (ONE-DAY 86400))
               (P (PERIODIC-AR-CLEANUP 24)
                  (PERIODIC-AR-REVIEW)))
;;
;; The trickle stuff
;;
```

```
        (FUNCTIONS RESTART-TRICKLES START-EMULATOR-TRICKLES START-NOTECARDS-TRICKLES START-LISPCORE-TRICKLES
            FILE-SERVER-UPP)
        (FNS |Smart-Identifier| |Smart-Trickle| |TrickleProcessLogfile|)
        (DECLARE\: DONTEVAL@LOAD DOCOPY (FILES (SYSLOAD)
                                              PROMPTREMINDERS COPYFILES)
            (INITVARS (*FUJI-TRICKLE-MAIL-ADDRESS* "AR-MANAGERS:MV:envos, Shunko Inoue:KSPA:Fuji Xerox"))

        ;; Start the Trickle Stuff

        (P (START-EMULATOR-TRICKLES)
            (START-LISPCORE-TRICKLES)
            (START-NOTECARDS-TRICKLES)))))
```

;; Functions for helping do the routine chores of AR Administration. Set up to run things automatically.

```
(RPAQQ IDLE.PROFILE (TIMEOUT NIL FORGET NIL ALLOWED.LOGINS NIL DISPLAYFN (LAMBDA (W)
                                                            (IDLE.BOUNCING.BOX W "Auto AR
                                                                Cleanup Machine")))))
```

(RPAQQ **IDLE.TIMEOUT** NIL)

(RPAQQ **IDLE.DEFAULTMODE** UNLOCKED)

(ADDTOVAR **INITIALSLST** (SYBALSKY |John| |jds|))

(RPAQQ **LOGINHOST/DIR** |{AR:MV:ENVOS}<LISPARS>ADMIN>|)

(RPAQQ **SYSOUTGAG** T)

(RPAQQ **PROMPT#FLG** T)

(RPAQQ **DEFAULTREGISTRY** |envos|)

(RPAQQ **CH.DEFAULT.DOMAIN** MV)

(RPAQQ **CH.DEFAULT.ORGANIZATION** ENVOS)

```
(RPAQQ DIRECTORIES ("{DSK}/USR/LOCAL/LDE/LISPCORE/SOURCES/" "{DSK}/USR/LOCAL/LDE/LISPCORE/LIBRARY/"
                    "{DSK}/USR/LOCAL/LDE/LISPCORE/INTERNAL/LIBRARY/" "{DSK}/USR/LOCAL/LDE/LISPUSERS/"
                    "{DSK}<usr>local>lde>rooms>sources>" "{DSK}<usr>local>lde>rooms>users-src>" {DSK}))
```

```
(RPAQQ DISPLAYFONTDIRECTORIES ("{DSK}/usr/local/lde/fonts/display/publishing/"
                               "{DSK}/usr/local/lde/fonts/display/presentation/"
                               "{DSK}/usr/local/lde/lispcore/XeroxPrivate/Fonts"
                               "{DSK}/usr/local/lde/fonts/display/chinese/"
                               "{DSK}/usr/local/lde/fonts/display/JIS1/"
                               "{DSK}/usr/local/lde/fonts/display/JIS2/"
                               "{DSK}/usr/local/lde/fonts/display/miscellaneous/"
                               "{DSK}/usr/local/lde/fonts/display/printwheel/"))
```

(RPAQQ **DISPLAYFONTEXTENSIONS** (AC DISPLAYFONT STRIKE))

```
(RPAQQ INTERPRESSFONTDIRECTORIES ("{DSK}/usr/local/lde/fonts/interpress/publishing/"
                                  "{DSK}/usr/local/lde/fonts/interpress/presentation/"
                                  "{DSK}/usr/local/lde/lispcore/XeroxPrivate/Fonts"
                                  "{DSK}/usr/local/lde/fonts/interpress/chinese/"
                                  "{DSK}/usr/local/lde/fonts/interpress/JIS1/"
                                  "{DSK}/usr/local/lde/fonts/interpress/JIS2/"
                                  "{DSK}/usr/local/lde/fonts/interpress/miscellaneous/"
                                  "{DSK}/usr/local/lde/fonts/interpress/printwheel/"))
```

(RPAQQ **DEFAULTPRINTINGHOST** ("Twister:" "Tremor:"))

;;
;; Tools to load
;;

(DECLARE\: DONTEVAL@LOAD DOCOPY

(FILESLOAD PROMPTREMINDERS CROCK WHO-LINE FILEWATCH)

(LAFITEMODE 'NS)

(LOGOW "AR & Trickle")

(CROCK (CREATEREGION 900 800 100 100))
)

;;
;; The AR stuff
;;
;; Every other day AR.CLEANUP to create new summaries &c.
;;

```
(DEFINEQ
```

(**PERIODIC-AR-CLEANUP**
```
  (LAMBDA (HRS-BETWEEN-CLEANUPS START-TIME)                              ; Edited  7-Dec-88 04:49 by kk

    ;; Schedule regular AR Cleanups to happen at 04:00 daily (or at whatever start time is specified)

    (LET* ((1AM-TODAY (IDATE (CONCAT (SUBSTRING (DATE)
                                                1 10)
                                        "00:01:00")))
            (NEXT-START (COND
                            ((>= (IDATE)
                                 1AM-TODAY)
                             (+ 1AM-TODAY 86400))
                            (T 1AM-TODAY))))
          (SETREMINDER '|AR-Cleanup| (TIMES HRS-BETWEEN-CLEANUPS 3600)
                    '(ADD.PROCESS '(DO-PERIODIC-AR-CLEANUP))
                    (OR START-TIME (GDATE NEXT-START))
                    (+ (IDATE)
                       (CL:* 365 86400))))))
```

(**DO-PERIODIC-AR-CLEANUP**
```
  (LAMBDA NIL                                                           ; Edited 27-Sep-90 15:30 by gadener

    ;; Do the actual work for periodic AR cleanups:  Do the cleanup, then delete the excess files on the summary subdirectory.

    (DRIBBLE "{LPT}")
    (PRINTOUT T ";;;" T)
    (PAGEHEIGHT 0)
    (WINDOWPROP (PROCESS.WINDOW (THIS.PROCESS))
            'PAGEFULLFN
            'NILL)
    (PRINTOUT T ";;; * * * * * * * * * * * * * * * * * * * * * * * * *" T ";;; * * * * * * * * * * * * * * * * *
            * * * * * * * *" T ";;;" T ";;;              A R   C L E A N U P" T ";;;" T ";;; Started at " (DATE)
        T)
    (PRINTOUT T ";;; Deleting {DSK}AR.INDEX ... ")
    (AND (> (LENGTH (DIRECTORY '|{dsk}AR.INDEX|))
            1)
         (DELFILE '|{dsk}AR.INDEX|))
    (PRINTOUT T "done." T ";;; Starting AR.CLEANUP" T)
    (AR.CLEANUP T '{DSK} T '{DSK})
    (PRINTOUT T T T ";;; AR.CLEANUP finished at " (DATE)
        T ";;; Deleting .TXT & .IP files ..." T)
    (DIRECTORY "{AR:MV:ENVOS}<LISPARS>SUMMARIES>*.TXT;L" '(DELETE))
    (DIRECTORY "{AR:MV:ENVOS}<LISPARS>SUMMARIES>*.IP;L" '(DELETE))
    (AND (> (LENGTH (DIRECTORY "{AR:MV:ENVOS}<LISPARS>AR.INDEX"))
            2)
         (PRINTOUT T ";;; Deleting AR.INDEX ... " (DELFILE "{AR:MV:ENVOS}<LISPARS>AR.INDEX")
             T))
    (PRINTOUT T ";;; Setting cleanup to run again at " (GDATE (+ ONE-DAY (IDATE (CONCAT (SUBSTRING (DATE)
                                                                                                1 10)
                                                                                "00:01:00"))))
        "." T)
    (REMINDER.NEXTREMINDDATE '|AR-Cleanup| (GDATE (+ ONE-DAY (IDATE (CONCAT (SUBSTRING (DATE)
                                                                                    1 10)
                                                                            "00:01:00")))))
    (DRIBBLE)))
)
```

```
;; Weekly creation of the list of ARs to review.

(DEFINEQ
```

(**PERIODIC-AR-REVIEW**
```
  (LAMBDA (INITIAL-DELAY)                                                ; Edited 27-Sep-90 15:46 by gadener

    ;; Schedule regular AR Reviews to happen at 07:00 every Monday

    (LET* ((7AM-TODAY (IDATE (CONCAT (SUBSTRING (DATE)
                                                1 10)
                                        "00:07:00")))
            (NEXT-START (COND
                            ((>= (IDATE)
                                 7AM-TODAY)
                             (+ 7AM-TODAY 86400))
                            (T 7AM-TODAY))))
          (SETREMINDER '|AR-Review| ONE-WEEK '(ADD.PROCESS '(DO-WEEKLY-AR-REVIEW))
                    (COND
                        (INITIAL-DELAY INITIAL-DELAY)
                        (T (GDATE NEXT-START)))
                    (+ (IDATE)
                       (CL:* 365 86400))))))
```

(**DO-WEEKLY-AR-REVIEW**
```
  (LAMBDA NIL                                                           ; Edited 28-Sep-90 13:49 by gadener
    (LET ((LAST-REVIEWED-AR (CAR (READFILE "{AR:MV:ENVOS}<LispARs>ADMIN>LAST-REVIEWED-AR")))
          LAST-FOUND-AR MESSAGE)
```

```
        (PRINTOUT T ";;;" T)
        (DRIBBLE "{LPT}")
        (PAGEHEIGHT 0)
        (WINDOWPROP (PROCESS.WINDOW (THIS.PROCESS))
               'PAGEFULLFN
               'NILL)
        (PRINTOUT T T ";;; * * * * * * * * * * * * * * * * * * * * * * * * *" T ";;; * * * * * * * * * * * * *
               * * * * * * * * * * * * *" T ";;;             A R   R E V I E W" T ";;;" T ";;; Starting at " (DATE)
               T)
        (SETQ LAST-FOUND-AR LAST-REVIEWED-AR)
        (|for| TRIAL-AR |from| LAST-REVIEWED-AR |while| (<= TRIAL-AR (+ LAST-FOUND-AR 50))
           |when| (INFILEP (PACKFILENAME.STRING 'NAME TRIAL-AR 'BODY "{AR:MV:ENVOS}<LISPARS>FOO.AR"))
           |do| (SETQ LAST-FOUND-AR TRIAL-AR))
        (LAFITEMODE 'NS)
        (SETQ MESSAGE (CONCAT "Subject:  ARs to Review
                         " "To:  AR-Managers:MV:envos

                         " (CL:FORMAT NIL "Printing for review ARs ~d-~d.  " (ADD1 LAST-REVIEWED-AR)
                                 LAST-FOUND-AR)
                         "John will be around with them...." "

                         --The AR Daemon
                         "))
        (LAFITE.SENDMESSAGE (OPENTEXTSTREAM MESSAGE))
        (AR.HARDCOPY (|for| I |from| (ADD1 LAST-REVIEWED-AR) |to| LAST-FOUND-AR |collect| I))
        (CL:WITH-OPEN-FILE (NEW-AR "{AR:MV:ENVOS}<LispARs>ADMIN>LAST-REVIEWED-AR" :DIRECTION :OUTPUT :IF-EXISTS
                           :NEW-VERSION)
               (PRINT LAST-FOUND-AR NEW-AR))
        (PRINTOUT T ";;; AR Review copies printed at" (DATE)
               "." T)
        ;; Set up to run next week at the same time:

        (REMINDER.NEXTREMINDDATE '|AR-Review| (GDATE (+ STARTING-MONDAY (CL:* ONE-WEEK (CL:CEILING
                                                           (- (IDATE)
                                                             STARTING-MONDAY)
                                                           ONE-WEEK)))))))))
)

(DEFINEQ
```

(**MY-FLOOR**
```
  (LAMBDA (NUMBER DIVISOR)
    (CL:* DIVISOR (CL:FLOOR NUMBER DIVISOR))))
)

(DECLARE\: DONTEVAL@LOAD DOCOPY

(FILESLOAD ARCLEANUP ARHACK)
```

(RPAQ **STARTING-MONDAY** (IDATE "5-DEC-88 07:00:00"))

(RPAQQ **ONE-WEEK** 604800)

(RPAQQ **ONE-DAY** 86400)

(**PERIODIC-AR-CLEANUP** 24)

(**PERIODIC-AR-REVIEW**)
)

```
;;
;; The trickle stuff
;;
```

(CL:DEFUN **RESTART-TRICKLES** ()
```
    (FOR REMINDER IN (ACTIVEREMINDERNAMES) DO (COND
                                          ((NOT (OR (EQL REMINDER '|AR-Cleanup|)
                                                    (EQL REMINDER '|AR-Review|)))
                                           (DELDEF REMINDER 'REMINDERS))))
    ;; Got rid of the old reminder, now ready to restart everything again
```
    (**START-EMULATOR-TRICKLES**)
    (**START-LISPCORE-TRICKLES**)
    (**START-NOTECARDS-TRICKLES**)
```
    T)
```

(CL:DEFUN **START-EMULATOR-TRICKLES** ()
    (**DECLARE** (GLOBAL *FUJI-TRICKLE-MAIL-ADDRESS*))
```
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{DSK}<users>maiko>working>src>*" "{Fuusen:KSPA:Fuji
                     Xerox}<Venue>Emul>working>src>*" "{DSK}<python>medley-trickles/emulator/src-"
                    ,*FUJI-TRICKLE-MAIL-ADDRESS* "06:00" NIL NIL)
        (TIMES 60 (RAND 10 60)))
```

```
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{DSK}<users>maiko>working>inc>*" "{Fuusen:KSPA:Fuji
                                Xerox}<Venue>Emul>working>inc>*" "{DSK}<python>medley-trickles/emulator/inc-"
                                ,*FUJI-TRICKLE-MAIL-ADDRESS* "07:00" NIL NIL)
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{DSK}<users>maiko>working>bin>*" "{Fuusen:KSPA:Fuji
                                Xerox}<Venue>Emul>working>bin>*" "{DSK}<python>medley-trickles/emulator/bin-"
                                ,*FUJI-TRICKLE-MAIL-ADDRESS* "08:00" NIL NIL)
            (TIMES 60 (RAND 10 60)))))
```

(CL:DEFUN **START-NOTECARDS-TRICKLES** ()

;;;

;;;; Setup all the NoteCards Core Trickles

;;;

```
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{Pika:MV:ENVOS}<NoteCards>system>*"
                                "{DSK}/usr/local/lde/lispcore/notecards/system/*"
                                "{DSK}/python/medley-trickles/notecards/system-" "SA" "20:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{Pika:MV:ENVOS}<NoteCards>library>*"
                                "{DSK}/usr/local/lde/lispcore/notecards/library/*"
                                "{DSK}/python/medley-trickles/notecards/library-" "SA" "23:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{Pika:MV:ENVOS}<NoteCards>lispusers>*"
                                "{DSK}/usr/local/lde/lispcore/notecards/lispusers/*"
                                "{DSK}/python/medley-trickles/notecards/lispusers-" "SA" "23:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{Pika:MV:ENVOS}<NoteCards>internal>*"
                                "{DSK}/usr/local/lde/lispcore/notecards/internal/*"
                                "{DSK}/python/medley-trickles/notecards/internal-" "SA" "23:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{Pika:MV:ENVOS}<NoteCards>tools>*"
                                "{DSK}/usr/local/lde/lispcore/notecards/tools/*"
                                "{DSK}/python/medley-trickles/notecards/tools-" "SA" "23:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{Pika:MV:ENVOS}<NoteCards>patches>*"
                                "{DSK}/usr/local/lde/lispcore/notecards/patches/*"
                                "{DSK}/python/medley-trickles/notecards/patches-" "SA" "23:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))))
```

(CL:DEFUN **START-LISPCORE-TRICKLES** ()

;;;

;;;; Set up all the Lispcore trickles

;;;

```
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{PELE:MV:ENVOS}<LISPCORE>SOURCES>*"
                                "{dsk}/usr/local/lde/lispcore/sources/*"
                                "{DSK}/python/medley-trickles/lispcore/sources-" "SA" "00:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{PELE:MV:ENVOS}<LISPCORE>LIBRARY>*"
                                "{dsk}/usr/local/lde/lispcore/library/*"
                                "{DSK}/python/medley-trickles/lispcore/library-" "SA" "04:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{PELE:MV:ENVOS}<LISPCORE>INTERNAL>LIBRARY>*"
                                "{dsk}/usr/local/lde/lispcore/internal/library/*"
                                "{DSK}/python/medley-trickles/lispcore/internal-library-" "SA" "06:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))
    (SETREMINDER NIL NIL '(|Smart-Trickle| "{PELE:MV:ENVOS}<LISPCORE>INTERNAL>DOC>*"
                                "{DSK}/usr/local/lde/lispcore/internal/doc/*"
                                "{DSK}/python/medley-trickles/lispore/internal-doc-" "SA" "07:00" T
                                '(STATUS DATABASE))
            (TIMES 60 (RAND 10 60)))))
```

(CL:DEFUN **FILE-SERVER-UPP** (SERVER &KEY (MESSAGE NIL MESSAGEP)
                                            (TIMEOUT 30))

    ;; Checks for a certain period of time if server is up. It will try to find a directory specific for the type of server.

```
    (CL:BLOCK FILE-SERVER-UP-P
        (CL:UNLESS (NULL SERVER)
            (LET* ((DIRECTORYNAME (COND
```

```
                                        ((CL:SEARCH ":" SERVER)              ; This is an NS-server
                                         "DESKTOPS")
                                        ((CL:SEARCH "/N" 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| TIMEOUT |timerUnits| 'SECONDS |until| (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)))))))))

(DEFINEQ
```

(|Smart-Identifier|

```
  (LAMBDA (|Source| |Destination|)                                        ; Edited 22-Oct-90 17:27 by gadener

    ;; Will return a string composed of <Common-Directories>-<Source-Host>-to-(Destination-Host> if there is a common denominator between the
    ;; two, otherwise the Common-Directories will consist of the Source-Directory

    (LET* ((|Src-Host| (FILENAMEFIELD |Source| 'HOST))
           (|Dst-Host| (FILENAMEFIELD |Destination| 'HOST))
           (|Src-Dir| (STRING (U-CASE (FILENAMEFIELD |Source| 'DIRECTORY))))
           (|Dst-Dir| (STRING (U-CASE (FILENAMEFIELD |Destination| 'DIRECTORY))))
           (|Common-point| (CL:MISMATCH |Src-Dir| |Dst-Dir| :TEST 'STRING-EQUAL :FROM-END T)))
          (CONCAT (IF (OR (NULL |Common-point|)
                          (EQL (CL:LENGTH |Src-Dir|)
                               |Common-point|))
                   THEN ;; Could not find a common denominator, just return the Src-Dir

                        |Src-Dir|
                   ELSE ;; Just Return the common part, Should be the common directory part.

                        (CL:SUBSEQ |Src-Drir| |Common-point|))
                  "-" |Src-Host| " to " |Dst-Host|))))
```

(|Smart-Trickle|

```
  (LAMBDA (|Source| |Destination| |RootLogfileName| |MailAddress| |ScheduleAnotherOne| |DontReplaceOldVersions|
           |DontCopyExtensions| |Retries|)                               ; Edited 27-Nov-90 10:13 by automgr
```

;;; 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. If a server is playing hard to catch, FILE-SERVER-UP-P will try harder (wait longer) for
;;; each time around. .

```
    (LET ((RETRY-TIMES 4)                                              ; How many retries.
          (RETRY-PERIOD 15)                                           ; How many minutes between retries
          (TIMEOUT-PERIOD 60)                                         ; Give up on fileserver after multiples of this period (seconds)
          (|Identifier| (|Smart-Identifier| |Source| |Destination|))
          |DateString| |LogfileName|)
         (IF (NOT (AND (OR |Retries| (SETQ |Retries| RETRY-TIMES))
                                                              ; This is the first time around
                       (FILE-SERVER-UPP (STRING (FILENAMEFIELD |Source| 'HOST))
                               :TIMEOUT
                               (CL:* (+ (- RETRY-TIMES |Retries|)
                                        1)
                                     TIMEOUT-PERIOD))
                       (FILE-SERVER-UPP (STRING (FILENAMEFIELD |Destination| 'HOST))
                               :TIMEOUT
                               (CL:* (+ (- RETRY-TIMES |Retries|)
                                        1)
                                     TIMEOUT-PERIOD))))
          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 "Retry-" |Identifier| "-" (GENSYM)))
                                      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 PROMPTWINDOW "Trickle: Tried to connect to " (FILENAMEFIELD |Source|
                                                                                        'HOST)
                                   " and "
                                   (FILENAMEFIELD |Destination| 'HOST)
                                   " , but one or both, are down!" T (IF |ScheduleAnotherOne|
                                                                        THEN "Rescheduling another TRICKLE for
                                                                              tomorrow!"))
                      ;; Will reschedule another one for tomorrow
                      (IF |ScheduleAnotherOne|
                          THEN (SETREMINDER (MKATOM (CONCAT |Identifier| "-" (GENSYM)))
                                      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|))))
                      ;; Send a message about this
                      (IF (AND (\\CH.FIND.ORG.SERVER 'ENVOS T)
                               (GETD 'LAFITEMODE)
                               (GETD 'LAFITE.SENDMESSAGE)
                               (OR (EQL (LAFITEMODE)
                                        'NS)
                                   (MEMBER (LAFITEMODE)
                                           'NS)))
                          THEN (LAFITE.SENDMESSAGE (CONCAT "To: "AR-MANAGERS:MV:ENVOS" "
                                                           Subject: Trickle had trouble accessing servers." "
                                                           Format: Text

                                                           Trickle tried to access:

                                                           " (FILENAMEFIELD |Source| 'HOST)
                                                           " and "
                                                           (FILENAMEFIELD |Destination| 'HOST)
                                                           "

                                                           several times, but gave up at "
                                                           (DATE)
                                                           " ; one of the servers, or both, were down!" "

                                                           Source       : " |Source| "
                                                           Destination       : " |Destination|
                                                           (IF |ScheduleAnotherOne|
                                                               THEN "

                                                                    Rescheduling another TRICKLE for
                                                                    tomorrow!")
                                                           "

                                                           <-- Grettings from your friendly Trickler -->" "
                                                           "))))
          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| |Identifier|)
              ;; Will reschedule another trickle for tomorrow

              (IF |ScheduleAnotherOne|
                  THEN (SETREMINDER (MKATOM (CONCAT |Identifier| "-" (GENSYM)))
                            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| |Identifier|)                    ; Edited 22-Oct-90 18:20 by gadener
    (PROG ((|LogfileStream| (OPENTEXTSTREAM |LogfileName|))
           |EndsOfLines|
           (|NewLine| (CHARACTER (CHARCODE EOL)))
           |EndOfLine| |Deletions| |FailedP|)
      (IF (ZEROP (GETEOFPTR |LogfileStream|))
          THEN ;; Probably an error, people usually don't Trickle empty directories

               (IF |MailAddress|
                   THEN (TEDIT.INSERT |LogfileStream| (PACK* "Subject: (Error?) Trickle: " |Identifier|
                                                             |NewLine| "To: " |MailAddress| |NewLine|
                                                             |NewLine|)
                              1)
                        (LAFITE.SENDMESSAGE |LogfileStream|)
                        (CLOSEF |LogfileStream|)
                        (RETURN)))
      (TEDIT.SETSEL |LogfileStream| 1 1 'LEFT)
      (SETQ |EndsOfLines| (CONS 0 (WHILE (SETQ |EndOfLine| (TEDIT.FIND |LogfileStream| |NewLine|))
                                     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|
                   ;; This little sequence will always return T. It is used to check if copying of a file failed. It is put here to avoid having to
                   ;; go through the LogFile several times. Once an error was found, don't bother looking at the file any more.

                   (OR (NOT (NULL |FailedP|))
                       (IF (STRPOS "failed" (TEDIT.SEL.AS.STRING |LogfileStream|
                                                    (TEDIT.SETSEL |LogfileStream| |StartOfLine|
                                                          (ADD1 (IDIFFERENCE |EndOfLine| |StartOfLine|)))))
                           THEN (SETQ |FailedP| T))
                       T)
                   (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, extend it, otherwise add it to the collection. This collapsing makes the actual
            ;; deletion more efficient, since we expect to keep few of the lines.

            (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|)
                                                    )))))
      ;; 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 is not associated with a window.

      (NLSETQ (TEDIT.PUT |LogfileStream| |LogfileName| T T))
      ;; Construct a mail message and send it

      (IF |MailAddress|
          THEN (TEDIT.INSERT |LogfileStream|
                      (CONCAT "Subject: " (IF (ZEROP (GETEOFPTR |LogfileStream|))
```

```
                                                    THEN "(Empty) "
                                                    ELSE "")
                                (IF (NOT (NULL |FailedP|))
                                    THEN "(Failed) "
                                  ELSE "")
                                "Trickle: " |Identifier| |NewLine| "To: " |MailAddress| |NewLine| |NewLine|
                                (IF (NOT (NULL |FailedP|))
                                    THEN (CONCAT
                                              "*********************************************************"
                                              |NewLine| "WARNING! Some of the files where not copied. Please
                                              check this message," |NewLine| "or the log: " |LogfileName|
                                              |NewLine|
                                              "*********************************************************"
                                              |NewLine| |NewLine|)
                                      ELSE ""))
                              1)
                      (LAFITE.SENDMESSAGE |LogfileStream|))
            (CLOSEF |LogfileStream|)))))
)

(DECLARE\: DONTEVAL@LOAD DOCOPY

(FILESLOAD (SYSLOAD)
       PROMPTREMINDERS COPYFILES)

(RPAQ? *FUJI-TRICKLE-MAIL-ADDRESS* "AR-MANAGERS:MV:envos, Shunko Inoue:KSPA:Fuji Xerox")

(START-EMULATOR-TRICKLES)

(START-LISPCORE-TRICKLES)

(START-NOTECARDS-TRICKLES)
)

(PUTPROPS ARADMIN COPYRIGHT ("Venue" 1990))
```

FUNCTION INDEX

VARIABLE INDEX