

File created: 12-Sep-91 19:04:19 {PELE:MV:ENVOS}<LISPCORE>INTERNAL>LIBRARY>ARHACK.;3

changes to: (VARS ARHACKCOMS)
(FNS OBSOLETE.MANY.ARS)

previous date: 15-Jun-90 10:53:51 {PELE:MV:ENVOS}<LISPCORE>INTERNAL>LIBRARY>ARHACK.;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1984, 1985, 1986, 1987, 1988, 1989, 1990, 1991 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ ARHACKCOMS

```
( (COMS ;; FNS for finding things out about lots of ARs
  (FNS AR.GET.PLIST)
                                     ; Grab specified field from set of ARs:
  (FNS NAB.TEST.CASES)
                                     ; Find ARs edited before a given date:
  (FNS LAST-CHANGED-BEFORE) )
 (COMS ;; For nabbing AR information from tds
  (FNS AR.GET.SUBMITS.FROM.TDS AR.GET.FIXES.FROM.TDS AR.GET.ARS.FROM.TDS COLLECT-FIXES
    COLLECT-SUBMISSIONS) )
 (COMS ;; Bulk changes to the AR database (fixes, closing, etc.)
  ;; Function to mark lots of ARs fixed in one sweep.
  (FNS FIX.MANY.ARS FIX.NO.RELEASE.NOTE)
  ;; Function for making changes to lots of ARs in one sweep; less convenient than above fn:
  ;; the form of CHANGE.SPEC is a list like ((1 Status: Closed Attn: "") (234 Attn: Me))
  (FNS CHANGE.MANY.ARS)
  ;; Function for closing lots of ARs, to mark lots of ARs obsolete.
  (FNS CLOSE.MANY.ARS OBSOLETE.MANY.ARS)
  ;; Functions for counting info about ARs
  (FNS LIST.NUM.STATUS.PRIORITY COUNT-ARS COUNT-BY-WEEK BREAKOUT-ARS-BY-TYPE) )
 (COMS ;; Functions for making a summary of all feature requests or hopefully-type ARs.
  (FNS FEATURE.SUMMARY HOPE.SUMMARY OPEN.SUMMARY FIXED.SUMMARY) )
 [COMS ;; Function for making lyric relevant summaries of Absolutely bugs, and a listing of all open ARs.
  (FNS ABSOLUTELY.SUMMARIES ABS.SUMMARY)
  (INITVARS (AR.ABSOLUTELY.HACKER.NAMES ' (Bane Biggs Burton Charnley Cude Daniels Fischer Kelley
                                           Murage Pavel Pedersen Shih Snow Sye SCPeters
                                           RMRichardson Sybalsky vanMelle Woz]
                                     ; Function for caching the ar index locally, and causing all later
                                     ; queries to use the local cache.
  (COMS (INITVARS (LOCAL-CACHE-ORIGINAL-AR-INDEX-NAME AR.INDEX.DEFAULT.FILE.NAME) )
        (FNS CACHE-ARINDEX) )
  (COMS ;; Fns for moving closed/obsolete/etc ar files from main directory to the <ClosedLispARs> directory.
        (FNS MAKE-AR-FILENAME MOVE-CLOSED-ARS) )
  (COMS ;; These names superseded by other functions on this file or on AREDDIT
        (FNS GET.NUMS.FROM.QUERY LANDSCAPE.QUERY.WINDOW HARDCOPY.ARS) )
  (COMS ;; Function for querying, and getting a list of the selected AR numbers. In effect, this is a programmatic interface to AR.QUERY.
        (FNS GET-SELECTED-AR-NUMBERS) )
  (COMS                                     ; Functions for making release notes & the like:
        ;; NOTES takes a list of items (AR# subj-string relnote-string) and prints them one to a line, with tab after AR# & new-line (not
        ;; new-para) before relnote into a TEdit.
        (FNS NOTES)
        (FNS KNOWN-BUG-LIST FIXED-BUG-LIST) ) ) )
```

;; FNS for finding things out about lots of ARs

(DEFINEQ

(AR.GET.PLIST

```
[LAMBDA (ARLIST ITEMS AS-STRINGS? SCRATCH.STREAM)
  ;; Edited 23-Feb-88 17:42 by bvm
  ;; Read ITEMS about each AR in ARLIST. RETURN a list of (AR# .items)...
  [OR SCRATCH.STREAM (SETQ SCRATCH.STREAM (OPENSTREAM ' {NODIRCORE} ' BOTH]
  (for NUM/OR/FILE in ARLIST bind MAP CONDITION SPEC STR
    collect (CL:MULTIPLE-VALUE-SETQ (MAP CONDITION)
                                     (AR.FETCH.AND.PARSE.AR NUM/OR/FILE SCRATCH.STREAM ITEMS) )
                                     ; Fetch desired fields into scratch stream, map shows where
  (CONS NUM/OR/FILE (if CONDITION
```

```

    then (CL:FORMAT PROMPTWINDOW "~%%Can't read AR ~A because: ~A" NUM/OR/FILE
          CONDITION)
          NIL
    else (for ITEMNAME in ITEMS collect (if (SETQ SPEC (CDR (ASSOC ITEMNAME MAP)))
      then
        ; Spec = (ptr length)
        (SETFILEPTR SCRATCH.STREAM
          (CAR SPEC))
        (SETQ STR (ALLOCSTRING (CADR SPEC)))
        (\BINS SCRATCH.STREAM
          (fetch (STRINGP BASE)
            of STR)
          0
          (CADR SPEC))
        (if AS-STRINGS?
          then STR
          else (MKATOM STR))
      )
    )

```

)

;; Grab specified field from set of ARs:

(DEFINEQ

(NAB.TEST.CASES

; Edited 23-Feb-88 17:46 by bvm

```

[LAMBDA (FIELD-TO-GET NUMS.OR.QFORMWINDOW)
  (LET ((NUMS (OR (LISTP NUMS.OR.QFORMWINDOW)
    (AR.NUMS.FROM.QUERY NUMS.OR.QFORMWINDOW)))
    (SCRATCH (OPENSTREAM '{NODIRCORE} 'BOTH))
    (ITEMS (LIST FIELD-TO-GET 'Subject%:))
    TEXT INFO)
    (printout T "will examine AR numbers: " NUMS T T)
    (for NUM in NUMS when (AND [SETQ TEXT (CADR (SETQ INFO (CAR (AR.GET.PLIST (LIST NUM)
      ITEMS T SCRATCH]
      (> (NCHARS TEXT)
        0))
    do (printout T .I5 NUM %,,, (CADDR INFO)
      T FIELD-TO-GET %,,, TEXT T T))

```

)

;; Find ARs edited before a given date:

(DEFINEQ

(LAST-CHANGED-BEFORE

; Edited 23-Feb-88 17:18 by bvm

```

;; Given a list of AR numbers, ARLIST, returns a list of those ARs that were last edited before the date specified in DATE-STRING (which must be
;; acceptable to IDATE).

```

```

(for TRIPLE in (AR.GET.PLIST ARLIST ' (Edit-Date%: Date%:)
  T)
  bind (DATE _ (IDATE DATE-STRING)) when (<= (IDATE (if (> (NCHARS (CADR TRIPLE))
    0)
    then
      ; Use edit date
      (CADR TRIPLE)
    else
      ; Not edited, so use submit date
      (CADDR TRIPLE)))
    DATE)
  collect (CAR TRIPLE])

```

)

;; For nabbing AR information from tds

(DEFINEQ

(AR.GET.SUBMITS.FROM.TDS

; Edited 23-Feb-88 18:15 by bvm

```

[LAMBDA (FILENAME)
  ;; Gather the list of ARs that have been submitted from the "Tool Driver Script" file, where AREDIT makes note of edits that people make.
  (AR.GET.ARS.FROM.TDS FILENAME 'SUBMIT])

```

(AR.GET.FIXES.FROM.TDS

; Edited 23-Feb-88 18:16 by bvm

```

[LAMBDA (FILENAME)
  ;; Gather the list of ARs from the "Tool Driver Script" file whose status has changed away from Open
  (AR.GET.ARS.FROM.TDS FILENAME 'EDIT (FUNCTION (LAMBDA (AR# INFO)
    (OR (STRPOS "->Fixed" INFO)
      (STRPOS "->Declined" INFO)
      (STRPOS "->Superseded" INFO)
      (STRPOS "->Obsolete" INFO))

```

(AR.GET.ARS.FROM.TDS

```

[LAMBDA (FILENAME TYPE PREDFN) ; Edited 23-Feb-88 18:13 by bvm
;; Reads the TDS file and returns a list (AR# changelInfo) for all ars of TYPE (EDIT, SUBMIT or NIL for either). If PREDFN is supplied, returns only
;; those for which (predfn AR# changelInfo) is true
(LET ((FILE (OPENSTREAM (OR FILENAME AR.INFO.FILE.NAME)
' INPUT
' OLD))
(*READTABLE* FILERDTBL))
(CL:UNWIND-PROTECT
(bind (SEARCHSTR (CONCAT " -- (" (OR TYPE "")))
AR# INFO while (FILEPOS SEARCHSTR FILE NIL NIL NIL T UPPERCASEARRAY)
when (PROGN ; Line looks like -- (EDIT AR# name date etc), where for old
; aredit there were parens around the other info
(if (NULL TYPE) ; Skip edit type
then
(SKREAD FILE))
(SETQ AR# (CL:READ FILE))
(SETQ INFO (if (EQ (CHCON1 (SETQ INFO (CL:READ-LINE FILE)))
(CHARCODE "("))
then ; Old kind
(SUBSTRING INFO 2 -3)
else ; New kind, just drop the last paren
(SUBSTRING INFO 1 -2)))
(OR (NULL PREDFN)
(CL:FUNCALL PREDFN AR# INFO))
collect (LIST AR# INFO))
(CLOSEF FILE)))

```

(COLLECT-FIXES

```

[LAMBDA NIL ; Edited 23-Apr-87 07:22 by jds
(for FILE in (DIRECTORY '{ERIS}<LISPARS>*.TDS-PROCESSED) join (AR.GET.FIXES.FROM.TDS FILE))

```

(COLLECT-SUBMISSIONS

```

[LAMBDA NIL ; Edited 23-Apr-87 07:21 by jds
(for FILE in (DIRECTORY '{ERIS}<LISPARS>*.TDS-PROCESSED) join (AR.GET.SUBMITS.FROM.TDS FILE))

```

```

)

```

```

;; Bulk changes to the AR database (fixes, closing, etc.)

```

```

;; Function to mark lots of ARs fixed in one sweep.

```

```

(DEFINEQ

```

(FIX.MANY.ARS

```

[LAMBDA (ARLIST STATUS NO.RELEASE.NOTE) ; Edited 5-Aug-88 11:04 by bvm
;; ARLIST is a list of AR numbers. Each AR on the list will be marked FIXED, or will be marked with the marking STATUS. Also moves the ATTN
;; field into the ASSIGNED-TO field and clears the ATTN field. If NO.RELEASE.NOTE is true, fills release note field with "none required"
(if [AND STATUS (NOT (MEMBER STATUS '(New Open Fixed Closed Declined Superseded Incomplete Obsolete Wish)
then (\ILLEGAL.ARG STATUS))
(for AR# in ARLIST bind (ARFORM _ (AR.SELECT.WINDOW "Select AR form"))
ARMENU BAD.ARS
first (SETQ ARMENU (AR.GET.MENU.FROM.MAIN.WINDOW ARFORM))
do
(printout T "Fixing " AR# T)
(COND
([NOT
(NLSETQ
(LET (NEWVALUE)
(AR.FORM.PROGRAMMATIC.GET ARMENU AR#)
[SETQ NEWVALUE (CL:STRING-TRIM ' (#\Space)
(AR.GET.BUTTON.FIELD.AS.TEXT ARFORM 'Attn%:)]
; Attn field, trimmed of spaces
[AR.REPLACE.FILL.INS
ARFORM
'([Status%: , (OR STATUS 'Fixed)
(Attn%: " ")
,@[if (> (NCHARS NEWVALUE)
0)
then
; Only change the ASSIGNED-TO field if the ATTN field had
; something in it.
'((|Assigned To:| ,NEWVALUE]
,@(if (AND NO.RELEASE.NOTE (NOT (> [NCHARS (CL:STRING-TRIM ' (#\Space)
(AR.GET.BUTTON.FIELD.AS.TEXT ARFORM
'|Release Note:|]
0)))
then ; Say none needed if there isn't a note yet
'((|Release Note:| "None required")
(AR.FORM.PROGRAMMATIC.PUT ARMENU]
(printout T "error reading AR!!!" T)
(push BAD.ARS AR#))
(TEDIT.STREAMCHANGEDP (TEXTSTREAM ARFORM)
T)

```

```

finally (if BAD.ARS
         then (printout T "bad ARs: " BAD.ARS T))

```

(FIX.NO.RELEASE.NOTE

[LAMBDA (ARLIST STATUS)

; Edited 23-Feb-88 18:42 by bvm

```

;; ARLIST is a list of AR numbers. Each AR on the list will be marked FIXED, and a note needing no release note.. Also moves the ATTN field into
;; the ASSIGNED-TO field and clears the ATTN field.

```

```

(FIX.MANY.ARS ARLIST STATUS T))

```

)

```

;; Function for making changes to lots of ARs in one sweep; less convenient than above fn:

```

```

;; the form of CHANGE.SPEC is a list like ((1 Status: Closed Attn: "") (234 Attn: Me))

```

(DEFINEQ

(CHANGE.MANY.ARS

[LAMBDA (CHANGE.SPEC)

; Edited 5-Aug-88 13:14 by bvm

```

;; the form of CHANGE.SPEC is a list like ((1 Status: Closed Attn: "") (234 Attn: Me))

```

```

(LET* ((ARFORM (AR.SELECT.WINDOW "Select AR form"))
       (ARMENU (AR.GET.MENU.FROM.MAIN.WINDOW ARFORM))
       (NCHANGES (LENGTH CHANGE.SPEC))
       BAD.ARS)
  (for CHANGE in CHANGE.SPEC as CNT from 1
    do (printout T "doing change: " CHANGE " (" .11 CNT "/" .11 NCHANGES ")" T)
        (if [NOT (NLSETQ (PROGN (AR.FORM.PROGRAMMATIC.GET ARMENU (CAR CHANGE))
                                [AR.REPLACE.FILL.INS ARFORM (for X on (CDR CHANGE)
                                                                by (CDDR X)
                                                                collect (LIST (CAR X)
                                                                (CADR X)]
                                (AR.FORM.PROGRAMMATIC.PUT ARMENU]
                                then (printout T "error reading AR!!!" T)
                                (push BAD.ARS CHANGE))
        (TEDIT.STREAMCHANGEDP (TEXTSTREAM ARFORM)
                               T)
    finally (printout T "bad ARs: " BAD.ARS T))

```

)

```

;; Function for closing lots of ARs, to mark lots of ARs obsolete.

```

(DEFINEQ

(CLOSE.MANY.ARS

[LAMBDA (ARLIST)

; Edited 5-Aug-88 11:07 by bvm

```

;; ARLIST is a list of AR numbers. Each AR on the list will be marked CLOSED.

```

```

(DECLARE (SPECVARS AR.NO.MESSAGE.FLG))
(LET* ((AR.NO.MESSAGE.FLG T)
       (ARFORM (AR.SELECT.WINDOW "Select AR form"))
       (ARMENU (AR.GET.MENU.FROM.MAIN.WINDOW ARFORM))
       BAD.ARS)
  (for AR# in ARLIST as CNT from 1 do (printout T "Closing " AR# T)
      (COND
        ((NLSETQ (PROGN (AR.FORM.PROGRAMMATIC.GET ARMENU AR#)
                        (COND
                          ((STRING-EQUAL "Fixed" (
                                AR.GET.BUTTON.FIELD.AS.TEXT
                                ARFORM
                                'Status%:))
                          [AR.REPLACE.FILL.INS ARFORM
                            `((Status%: Closed]
                          (AR.FORM.PROGRAMMATIC.PUT ARMENU))
                          (T (PRINTOUT T "AR not Fixed first: " AR# T)
                             (push BAD.ARS AR#)))
                        NIL))
          (T (printout T "error reading AR!!!" T)
              (push BAD.ARS AR#)))
        (TEDIT.STREAMCHANGEDP (TEXTSTREAM ARFORM)
                               T)
    finally (printout T "ARs not closed: " (COND
                                             (BAD.ARS)
                                             (T "None."))
               T))

```

(OBSOLETE.MANY.ARS

[LAMBDA (ARLIST)

; Edited 12-Sep-91 18:57 by jds

```

;; ARLIST is a list of AR numbers. Each AR on the list will be marked OBSOLETE, and any ATTN will be removed..

```

```

(DECLARE (SPECVARS AR.NO.MESSAGE.FLG))
(LET* ((AR.NO.MESSAGE.FLG T)

```

```

    (ARFORM (AR.SELECT.WINDOW "Select AR form"))
    (ARMENU (AR.GET.MENU.FROM.MAIN.WINDOW ARFORM))
    BAD.ARS)
  (for AR# in ARLIST as CNT from 1 do (printout T "Closing " AR# T)
    (COND
      ((NLSETQ (PROGN (AR.FORM.PROGRAMMATIC.GET ARMENU AR#)
        [AR.REPLACE.FILL.INS ARFORM `((Status%: Obsolete)
          (Attn%: ""])
        (AR.FORM.PROGRAMMATIC.PUT ARMENU)
        NIL))
        NIL)
      (T (printout T "error reading AR!!!" T)
        (push BAD.ARS AR#)))
    (TEDIT.STREAMCHANGEDP (TEXTSTREAM ARFORM)
      T)
    finally (printout T "ARs not obsoleted: " (COND
      (BAD.ARS)
      (T "None."))
      T))
  )

```

;; Functions for counting info about ARs

(DEFINEQ

(LIST.NUM.STATUS.PRIORITY

[LAMBDA (ARLIST FIELDS)

; Edited 23-Feb-88 18:51 by bvm

;; ARLIST is a list of AR numbers. Returns a list of entries of the form (# fieldValue fieldValue ...). where each field is one of the names in FIELDS.

```

  (AR.GET.PLIST ARLIST (OR FIELDS '(Status%: Priority%: |Problem Type:| System%: Subsystem%:))
    T))

```

(COUNT-ARS

[LAMBDA (ARLIST)

; Edited 23-Apr-87 08:10 by jds

```

  (PRINTOUT T T T "For Documentation:" T T)

```

```

  (bind FIXED NEW OPEN OTHER ABS HOPE F/A F/H F/O O/A O/H O/O first
    (SETQ FIXED 0)
    (SETQ NEW 0)
    (SETQ OPEN 0)
    (SETQ OTHER 0)
    (SETQ ABS 0)
    (SETQ HOPE 0)
    (SETQ F/A 0)
    (SETQ F/H 0)
    (SETQ F/O 0)
    (SETQ O/A 0)
    (SETQ O/H 0)
    (SETQ O/O 0)

```

```

  for AR in ARLIST when (MEMBER 'Documentation AR) do (SELECTQ (CADR AR)
    ((New Open)
      (add NEW 1)
      (SELECTQ (CADDR AR)
        (Absolutely (add ABS 1)
          (add O/A 1))
        (Hopefully (add HOPE 1)
          (add O/H 1))
        (PROGN (add OTHER 1)
          (add O/O 1))))
    ((Fixed Declined Obsolete Incomplete Superseded)
      (add FIXED 1)
      (SELECTQ (CADDR AR)
        (Absolutely (add ABS 1)
          (add F/A 1))
        (Hopefully (add HOPE 1)
          (add F/H 1))
        (PROGN (add OTHER 1)
          (add F/O 1))))
    (SELECTQ (CADDR AR)
      (Absolutely (add ABS 1))
      (Hopefully (add HOPE 1))
      (add OTHER 1)))

```

```

  finally (PRINTOUT T "      NEW      FIXED" T "ABS  " O/A "      " F/A T "HOPE " O/H "      " F/H T "OTHR " O/O
    "      " F/O T T))

```

;; Now FEATURES

```

  (PRINTOUT T T T T "For FEATURES:" T T)

```

```

  (bind FIXED NEW OPEN OTHER ABS HOPE F/A F/H F/O O/A O/H O/O first
    (SETQ FIXED 0)
    (SETQ NEW 0)
    (SETQ OPEN 0)
    (SETQ OTHER 0)
    (SETQ ABS 0)
    (SETQ HOPE 0)
    (SETQ F/A 0)
    (SETQ F/H 0)
    (SETQ F/O 0)

```



```

(EQ 'Performance (CL:FOURTH AR]
do (SELECTQ (CADDR AR)
  ((New Open)
    (add NEW 1)
    (SELECTQ (CADDR AR)
      (Absolutely (add ABS 1)
        (add O/A 1))
      (Hopefully (add HOPE 1)
        (add O/H 1))
      (PROGN (add OTHER 1)
        (add O/O 1))))
    ((Fixed Declined Obsolete Incomplete Superseded)
      (add FIXED 1)
      (SELECTQ (CADDR AR)
        (Absolutely (add ABS 1)
          (add F/A 1))
        (Hopefully (add HOPE 1)
          (add F/H 1))
        (PROGN (add OTHER 1)
          (add F/O 1))))
      (SELECTQ (CADDR AR)
        (Absolutely (add ABS 1))
        (Hopefully (add HOPE 1))
        (add OTHER 1)))
  finally (PRINTOUT T "      NEW      FIXED" T "ABS  " O/A  "      " F/A T "HOPE " O/H  "      " F/H T "OTHR " O/O
    "      " F/O T T])

```

(COUNT-BY-WEEK

; Edited 23-Apr-87 14:17 by jds

```

[LAMBDA NIL
  (LET ((SUBMITS BETASUBMITS)
    (FIXES BETAFIXES)
    (WASFIXED (SORT SUBMITTEDFIXED))
    SUB FIX SUBFIX SD)
    (for START from (IDATE "2-FEB-87 00:00:00") by 604800 as END from (IDATE "9-FEB-87 00:00:00")
      by 604800 while (OR SUBMITS FIXES)
      do (PRINTOUT T T T "Week of " (GDATE START)
        " - "
        (GDATE END)
        T T)
      [SETQ SUB (bind AR
        while [AND SUBMITS (IGEQ END (IDATE (CONCAT (CAR (NTH (CADDR (SETQ AR (CAR SUBMITS)))
          (IDIFFERENCE
            (LENGTH (CADDR AR))
            1)))
          " 00:00"]
        collect (CAR (pop SUBMITS])
      [SETQ FIX (bind AR while [AND FIXES (IGEQ END (IDATE (CONCAT [CADDR (CADDR (SETQ AR (CAR FIXES]
        " 00:00"]
        collect (CADDR (pop FIXES])
      (SETQ SUBFIX (for AR in SUB while WASFIXED when (COND
        ((NULL WASFIXED)
        NIL)
        ((IEQP AR (CAR WASFIXED))
        (pop WASFIXED)
        T)
        ((ILESSP AR (CAR WASFIXED))
        NIL)
        (T (HELP "Mismatch -- a sub-as-fixed ar never
          got submitted?"))))
        collect AR))
      (PRINTOUT T "      Submitted: " .I3 (LENGTH SUB)
        "
        ")
      (BREAKOUT-ARS-BY-TYPE SUB)
      (PRINTOUT T "Submitted&Fixed: " .I3 (LENGTH SUBFIX)
        "
        ")
      (BREAKOUT-ARS-BY-TYPE SUBFIX)
      (PRINTOUT T "      Newly Fixed: " .I3 (LENGTH FIX)
        "
        ")
      (BREAKOUT-ARS-BY-TYPE FIX)
      (PRINTOUT T T])

```

(BREAKOUT-ARS-BY-TYPE

; Edited 23-Feb-88 18:26 by bvm

```

[LAMBDA (ARLIST)
  (bind (OTHER _ 0)
    (ABS _ 0)
    (HOPE _ 0)
    (FEAT _ 0)
    (PERF _ 0)
    (DOCABS _ 0)
    (DOCHOPE _ 0)
    (DOCOTHER _ 0)
    (DUNNO _ 0)
    AR for AR# in ARLIST do (SETQ AR (ASSOC AR# BETAINFO))
    [COND
      ( (NULL AR)

```

```

      (add DUNNO 1))
    ( (MEMBER 'Documentation AR)
      (SELECTQ (CADDR AR)
        (Absolutely (add DOCABS 1))
        (Hopefully (add DOCHOPE 1))
        (add DOCOTHER 1)))
    ( (MEMBER 'Feature AR)
      (add FEAT 1))
    ( (MEMBER 'Performance AR)
      (add PERF 1))
    (T (SELECTQ (CADDR AR)
      (Absolutely (add ABS 1))
      (Hopefully (add HOPE 1))
      (add OTHER 1)
      finally (PRINTOUT T "BUGS:  Abs: " .I3 ABS "  Hope: " .I3 HOPE "  Other: " .I3 OTHER "  || DOC: Abs: "
        .I3 DOCABS "  Hope: " .I3 DOCHOPE "  Other: " .I3 DOCOTHER "  || Features: " .I3 FEAT "
        Perf: " PERF (COND
          ((ZEROP DUNNO)
            "")
          (T (CONCAT "  Can't Tell: " DUNNO)))
        T}))
  )

```

;; Functions for making a summary of all feature requests or hopefully-type ARs.

```
(DEFINEQ
```

(FEATURE.SUMMARY

```
[LAMBDA (INDEX-WINDOW SUMMARY.LOCAL.DIR)
```

; Edited 24-Feb-88 20:07 by bvm

;;; Make a summary of all extant feature-request ARs:

```

  (AR.QUERY.PRINT.AND.IP.FILE INDEX-WINDOW '[AND (OR (Status%: IS Open)
    (Status%: IS New))
    (|Problem Type:| IS Feature)
    (NOT (OR (System%: IS LOOPS)
      (System%: IS PCE)
      (System%: IS PROLOG)
      (System%: IS BusMaster)
      (Subsystem%: IS DEI]
    '(System%: Subsystem%: Priority%:)
    (CONCAT (OR SUMMARY.LOCAL.DIR '{DSK6})
      "FeatureSummary"]])

```

(HOPE.SUMMARY

```
[LAMBDA (INDEX-WINDOW SUMMARY.LOCAL.DIR FIELDS-TO-PRINT)
```

; Edited 24-Feb-88 20:15 by bvm

;;; Make a summary of all Lyric Absolutely ARs:

```

  (AR.QUERY.PRINT.AND.IP.FILE INDEX-WINDOW '[AND (OR (Status%: IS Open)
    (Status%: IS New))
    (Priority%: IS Hopefully)
    (NOT (OR (|Problem Type:| IS Feature)
      (|Problem Type:| IS Documentation)
      (System%: IS LOOPS)
      (System%: IS PCE)
      (System%: IS PROLOG)
      (System%: IS BusMaster)
      (Subsystem%: IS DEI]
    '(System%: Subsystem%: Priority%:)
    (CONCAT (OR SUMMARY.LOCAL.DIR '{DSK6})
      "HopeSummary")
    NIL FIELDS-TO-PRINT])

```

(OPEN.SUMMARY

```
[LAMBDA (SUMMARY.LOCAL.DIR)
```

; Edited 3-Feb-89 10:29 by jds

;;; Make a summary of all ARs that are Open.

```

  (AR.QUERY.PRINT.AND.IP.FILE NIL '[OR (Status%: IS New)
    (Status%: IS Open)
    (Status%: IS Open/Unreleased)
    (Status%: IS Incomplete)
    (Status%: IS Internal)
    (Status%: IS Wish))
    '(System%: Subsystem%: Priority%:)
    "OpenSummary"
    (OR SUMMARY.LOCAL.DIR '{DSK}))

```

(FIXED.SUMMARY

```
[LAMBDA (SUMMARY.LOCAL.DIR)
```

; Edited 24-Feb-88 20:28 by bvm

;;; Make a summary of all ARs that are FIXED


```

(LET [(LOCAL.AR.INDEX.NAME (AND SUMMARY.LOCAL.DIR (CONCAT SUMMARY.LOCAL.DIR 'AR.INDEX]
;; Copy the AR index to local disk if it's not there already.
[COND
  (LOCAL.AR.INDEX.NAME (COPYFILES AR.INDEX.DEFAULT.FILE.NAME LOCAL.AR.INDEX.NAME '>A]
  (AR.QUERY.PRINT.AND.IP.FILE LOCAL.AR.INDEX.NAME ' (AND (Status%: IS Fixed))
  NIL
  (PACKFILENAME.STRING 'NAME "FixedSummary" 'DIRECTORY SUMMARY.LOCAL.DIR)
  NIL
  ' ((Number%: 5)
    (Subject%: 60])
)

;; Function for making lyric relevant summaries of Absolutely bugs, and a listing of all open ARs.

(DEFINEQ

(ABSOLUTELY.SUMMARIES
  [LAMBDA (SUMMARY.LOCAL.DIR INDIVIDUAL-SUMMARIES? INDEX-ALREADY-COPIED? THESE-NAMES-ONLY)
    ; Edited 24-Feb-88 20:25 by bvm

;;; Create a summary for each developer, listing the ABSOLUTELY ARs in that person's name, and an "AllBugs" summary of Open/New ARs.
;; If INDIVIDUAL-SUMMARIES? (or THESE-NAMES-ONLY) then produce a summary for each developer listing his/her absolutelies. If
;; THESE-NAMES-ONLY, then only do it for those names.
(LET ([LOCAL.AR.INDEX.NAME (AND SUMMARY.LOCAL.DIR (CONCAT SUMMARY.LOCAL.DIR 'AR.INDEX]
  INDEX.WINDOW)
  (COND
    ((AND LOCAL.AR.INDEX.NAME (NOT INDEX-ALREADY-COPIED?))
      (printout T "copying old AR index to " LOCAL.AR.INDEX.NAME "...")
      (COPYFILES AR.INDEX.DEFAULT.FILE.NAME LOCAL.AR.INDEX.NAME '>A)
      (printout T "done" T))
    [SETQ INDEX.WINDOW (CREATEW ' (0 0 200 100]
    (AR.QFORM.CREATE LOCAL.AR.INDEX.NAME INDEX.WINDOW T)

;; Make the summary of all open/new ARs, for cleanup purposes:
[COND
  ((NOT THESE-NAMES-ONLY)
    (AR.QUERY.PRINT.AND.IP.FILE INDEX.WINDOW ' (AND (OR (Status%: IS Open)
      (Status%: IS New))
      ' (System%: Subsystem%: Status%:)
      (PACKFILENAME.STRING 'NAME "AllBugsSummary" 'DIRECTORY SUMMARY.LOCAL.DIR))

;; Make a summary of all Lyric Absolutely ARs:
(AR.QUERY.PRINT.AND.IP.FILE INDEX.WINDOW ' [AND (OR (Status%: IS Open)
      (Status%: IS New))
      (Priority%: IS Absolutely)
      (NOT (OR (|Problem Type:| IS Feature)
        (|Problem Type:| IS Documentation)
        (System%: IS LOOPS)
        (System%: IS PCE)
        (System%: IS PROLOG)
        (System%: IS BusMaster)
        (Subsystem%: IS DEI]
      ' (System%: Subsystem%: Status%:)
      (PACKFILENAME.STRING 'NAME "LyricAbsSummary" 'DIRECTORY SUMMARY.LOCAL.DIR]

;; Make summaries for everyone:
[COND
  ((OR THESE-NAMES-ONLY INDIVIDUAL-SUMMARIES?)
    (for HACKER.NAME in (OR THESE-NAMES-ONLY AR.ABSOLUTELY.HACKER.NAMES)
      do ;; Query on non-feature, non-doc absolutelies for this guy
        (AR.QUERY.PRINT.AND.IP.FILE INDEX.WINDOW
          '[AND (OR (Status%: IS Open)
            (Status%: IS New))
            (Priority%: IS Absolutely)
            [Attn%: HAS ,(SUBSTRING HACKER.NAME 1 (IMIN 4 (NCHARS HACKER.NAME]
            (NOT (OR (|Problem Type:| IS Feature)
              (|Problem Type:| IS Documentation)
              AR.CLEANUP.SORT.ORDER
              (PACKFILENAME.STRING 'NAME (CONCAT HACKER.NAME "Summary")
                'DIRECTORY SUMMARY.LOCAL.DIR))

          ;; Print the summary, but don't copy it anywhere (leave it on the local summary dir)

        ]
    ]

;; Close the query window we used for this process.
(CLOSEW INDEX.WINDOW])

```

(ABS.SUMMARY

[LAMBDA (INDEX-WINDOW SUMMARY.LOCAL.DIR)

; Edited 24-Feb-88 20:27 by bvm

;;; Make a summary of all Lyric Absolutely ARs:

```
(AR.QUERY.PRINT.AND.IP.FILE NIL '[AND (OR (Status%: IS Open)
                                           (Status%: IS New))
                                   (Priority%: IS Absolutely)
                                   (In/By%: HAS Motown)
                                   (NOT (OR (System%: IS LOOPS)
                                           (System%: IS PCE)
                                           (System%: IS PROLOG)
                                           (System%: IS BusMaster)
                                           (Subsystem%: IS DEI]
                                           ' (System%: Subsystem%: Status%:))
                                   (CONCAT (OR SUMMARY.LOCAL.DIR '{DSK6})
                                           "LyricAbsSummary"]])
)

(RPAQ? AR.ABSOLUTELY.HACKER.NAMES '(Bane Biggs Burton Charnley Cude Daniels Fischer Kelley Murage Pavel
                                   Pedersen Shih Snow Sye SCPeters RMRichardson Sybalsky vanMelle Woz))
```

;; Function for caching the ar index locally, and causing all later queries to use the local cache.

```
(RPAQ? LOCAL-CACHE-ORIGINAL-AR-INDEX-NAME AR.INDEX.DEFAULT.FILE.NAME)

(DEFINEQ
  (CACHE-ARINDEX
    [LAMBDA (LOCALNAME)
      ; Edited 1-Jul-87 10:39 by jds
      ;; Copy the ar index to a LOCALNAME cache spot, and redirect the local pointers so queries operate there.
      ;; CAUTION: This will cause you not to see new versions of the index, and you'll lose the cache if you reload AREDIT et al.
      (COPYFILES LOCAL-CACHE-ORIGINAL-AR-INDEX-NAME LOCALNAME) ; Copy the file
      (SETQ AR.INDEX.DEFAULT.FILE.NAME LOCALNAME))
  )
```

;; Fns for moving closed/obsolete/etc ar files from main directory to the <ClosedLispARs> directory.

```
(DEFINEQ
  (MAKE-AR-FILENAME
    [LAMBDA (DIR NUMBER)
      ; Edited 7-Feb-90 17:12 by jds
      (PACKFILENAME.STRING 'DIRECTORY DIR 'BODY (CL:FORMAT NIL "~5,'0D.AR" NUMBER))
  )

  (MOVE-CLOSED-ARS
    [LAMBDA (NUMBERS)
      (for AR in NUMBERS do (RENAMEFILE (MAKE-AR-FILENAME "{ERINYES}<LISPARS>" AR)
                                         (MAKE-AR-FILENAME "{ERINYES}<CLOSEDLISPARS>" AR))
      (PRINTOUT T "Moved AR file for AR " AR T])
  )
```

;; These names superseded by other functions on this file or on AREDIT

```
(DEFINEQ
  (GET.NUMS.FROM.QUERY
    [LAMBDA NIL
      ; Edited 18-Feb-88 12:43 by bvm
      ;; Gather the AR numbers listed in a query window, and return a list of them. Useful for getting AR numbers into Lisp for further processing.
      (AR.NUMS.FROM.QUERY)]
  )
```

```
(LANDSCAPE.QUERY.WINDOW
  [LAMBDA (LOCALTXTFILE LOCALIPFILE)
    ; Edited 23-Feb-88 17:51 by bvm
    ;; This function takes the names of two files. LOCALTXTFILE is the file which has been created by PRINTing to a Print File in a query window.
    ;; LOCALIPFILE is the file that you want to be created with landscaping.
    (AR.IP.FILE LOCALTXTFILE LOCALIPFILE)])
```

```
(HARDCOPY.ARS
  [LAMBDA (NUMS)
    ; Edited 23-Feb-88 18:27 by bvm
    (AR.HARDCOPY NUMS)]
)
```

;; Function for querying, and getting a list of the selected AR numbers. In effect, this is a programmatic interface to AR.QUERY.

```
(DEFINEQ
  (GET-SELECTED-AR-NUMBERS
    [LAMBDA (QUERY-SPEC SUMMARY.LOCAL.DIR)
      ; Edited 17-Jan-89 18:20 by jds
```

;;; Return a list of all the AR numbers matching QUERY-SPEC. If SUMMARY.LOCAL.DIR, then cache the AR.INDEX file there for this operation.

```
(LET ((LOCAL.AR.INDEX.NAME (COND
  (SUMMARY.LOCAL.DIR (PACK* SUMMARY.LOCAL.DIR 'AR.INDEX))
  (T NIL)))
  INDEX.WINDOW SELECTED-ARS)
  ;; Copy the AR index to local disk if it's not there already.
  [COND
    (LOCAL.AR.INDEX.NAME (COPYFILES AR.INDEX.DEFAULT.FILE.NAME LOCAL.AR.INDEX.NAME '>A]
  ;; Open the query window
  (SETQ INDEX.WINDOW (CREATEW (CREATEREGION 0 0 200 100)))
  (AR.QFORM.GROUP.CREATE LOCAL.AR.INDEX.NAME INDEX.WINDOW T)
  ;; Grab the ARs & print the summary:
  (AR.QUERY INDEX.WINDOW QUERY-SPEC)
  (SETQ SELECTED-ARS (AR.NUMS.FROM.QUERY INDEX.WINDOW))
  (CLOSEW INDEX.WINDOW)
  SELECTED-ARS])
)
```

;; Functions for making release notes & the like:

;; NOTES takes a list of items (AR# subj-string relnote-string) and prints them one to a line, with tab after AR# & new-line (not new-para) before relnote
 ;; into a TEdit.

(DEFINEQ

(NOTES

[LAMBDA (ARS TITLE)

; Edited 17-Jan-89 18:28 by jds

;; Given a list of items like (AR# Subject-string Rel-Note-String), print them into a TEdit document one to a paragraph, with a newline after the
 ;; subject. Intended for the collation of release notes.

```
(LET ((TS (OPENTEXTSTREAM "")))
  (TEDIT.SETSEL TS 1 0 'RIGHT)
  (AND TITLE (TEDIT.INSERT TS (CONCAT TITLE "

" )))
  (for AR-INFO in ARS do (TEDIT.INSERT TS (CONCAT (MKSTRING (CAR AR-INFO))
    " "
    (CADR AR-INFO)
    (CHARACTER (CHARCODE "##^M"))
    (CADDR AR-INFO)
    "
    " )))
  (TEDIT TS])
)
```

(DEFINEQ

(KNOWN-BUG-LIST

[LAMBDA (SUMMARY.LOCAL.DIR)

; Edited 17-Jan-89 18:33 by jds

```
(LET (ARS NOTES)
  (SETQ ARS (GET-SELECTED-AR-NUMBERS '[AND (OR (Status%: IS Open)
    (Status%: IS Open/Unreleased)
    (Status%: IS New))
    (Priority%: IS Hopefully)
    (NOT (OR (Problem Type: IS Feature)
      (Problem Type: IS Documentation)
      (System%: IS LOOPS)
      (System%: IS PCE)
      (System%: IS PROLOG)
      (System%: IS BusMaster)
      (Subsystem%: IS DEI]
    SUMMARY.LOCAL.DIR))
  (SETQ ARS (AR.GET.PLIST ARS '(Subject%: |Release Note:|)
    T))
  (SETQ NOTES (NOTES ARS (CONCAT "Known Bugs List as of " (DATE]))
```

(FIXED-BUG-LIST

[LAMBDA (SUMMARY.LOCAL.DIR)

; Edited 17-Jan-89 18:32 by jds

```
(LET (ARS NOTES)
  (SETQ ARS (GET-SELECTED-AR-NUMBERS '[AND (Status%: IS Fixed)
    (NOT (OR (System%: IS LOOPS)
      (System%: IS PCE)
      (System%: IS PROLOG)
      (System%: IS BusMaster)
      (Subsystem%: IS DEI]
    SUMMARY.LOCAL.DIR))
  (SETQ ARS (AR.GET.PLIST ARS '(Subject%: |Release Note:|)
    T))
  (SETQ NOTES (NOTES ARS (CONCAT "Fixed Bugs List as of " (DATE]))
```

{MEDLEY}<internal>ARHACK.;1

Page 12

)

(PUTPROPS **ARHACK COPYRIGHT** ("Venue & Xerox Corporation" 1984 1985 1986 1987 1988 1989 1990 1991))

FUNCTION INDEX

ABS.SUMMARY	9	COLLECT-SUBMISSIONS	3	HOPE.SUMMARY	8
ABSOLUTELY.SUMMARIES	9	COUNT-ARS	5	KNOWN-BUG-LIST	11
AR.GET.ARS.FROM.TDS	3	COUNT-BY-WEEK	7	LANDSCAPE.QUERY.WINDOW	10
AR.GET.FIXES.FROM.TDS	2	FEATURE.SUMMARY	8	LAST-CHANGED-BEFORE	2
AR.GET.PLIST	1	FIX.MANY.ARS	3	LIST.NUM.STATUS.PRIORITY	5
AR.GET.SUBMITS.FROM.TDS	2	FIX.NO.RELEASE.NOTE	4	MAKE-AR-FILENAME	10
BREAKOUT-ARS-BY-TYPE	7	FIXED-BUG-LIST	11	MOVE-CLOSED-ARS	10
CACHE-ARINDEX	10	FIXED.SUMMARY	8	NAB.TEST.CASES	2
CHANGE.MANY.ARS	4	GET-SELECTED-AR-NUMBERS	10	NOTES	11
CLOSE.MANY.ARS	4	GET.NUMS.FROM.QUERY	10	OBSOLETE.MANY.ARS	4
COLLECT-FIXES	3	HARDCOPY.ARS	10	OPEN.SUMMARY	8

VARIABLE INDEX

AR.ABSOLUTELY.HACKER.NAMES	10	LOCAL-CACHE-ORIGINAL-AR-INDEX-NAME	10
----------------------------------	----	--	----
