

File created: 1-Apr-92 18:07:47 {PELE:MV:ENVOS}<LISPCORE>INTERNAL>LIBRARY>ARINDEX.;4

changes to: (VARS ARINDEXCOMS)

previous date: 15-Jun-90 10:59:57 {PELE:MV:ENVOS}<LISPCORE>INTERNAL>LIBRARY>ARINDEX.;3

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1988, 1990, 1992 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ ARINDEXCOMS

(; Creating and updating the index. Separate file because AREDIT doesn't need this

```
(FNS AR.GATHER.NEW.AR.DATA AR.INDEX.CREATE AR.GET.ENUMERATED.FIELD.KEYS AR.INDEX.FIND.ENTRY.PTR
AR.INDEX.REWRITE.ENTRY.DATA AR.INDEX.REWRITE.FIELD.DATA AR.QFORM.FN.PRINT.INDEX AR.INDEX.PRINT
AR.QFORM.FN.UPDATE AR.INDEX.UPDATE)
(FNS AR.GET.FIELD.VAL.LENGTH AR.GET.FIELD.VAL.PTR AR.GET.FIELD.VAL.SHAPE AR.GET.ENTRY.NUM)
[INITVARS (AR.INDEX.DEFAULT.FIELDS '(Subject%: Source%: Date%: Submitter%: |Assigned To:| Attn%:
Status%: In/By%: |Problem Type:| Impact%: Difficulty%:
Frequency%: Priority%: System%: Subsystem%: Machine%: Disk%:
Lisp Version:| Source Files:| Microcode Version:|
Memory Size:| File Server:| Server Software Version:|
Edit-By%: Edit-Date%:)]
(DECLARE%: EVAL@COMPILE DONTCOPY (FILES (LOADCOMP
ARQUERY)
(GLOBALVARS AR.INDEX.DEFAULT.FIELDS)
(FUNCTIONS AR.ENTRY.PTR.TO.KEY.VAL.PTR ARSPECPUT)))
```

:: Creating and updating the index. Separate file because AREDIT doesn't need this

(DEFINEQ

(AR.GATHER.NEW.AR.DATA

[LAMBDA (FORMWINDOW AR.NUM.LIST AR.SCRATCH.FILE) ; Edited 21-Jul-88 15:08 by bvm

:: AR.NUM.DATA should be a sorted list of AR numbers. AR.GATHER.NEW.AR.DATA returns a list with elements of the form (<arnum> <arptr> .
:: <ar.scratch.assoc>)

```
(LET [(AR.NUM.DATA (for AR.NUM in AR.NUM.LIST bind START
collect (BLOCK)
(SETQ START (GETFILEPTR AR.SCRATCH.FILE))
(LIST* AR.NUM NIL (CL:MULTIPLE-VALUE-BIND (INDEX.INFO CONDITION)
(AR.FETCH.AND.PARSE.AR AR.NUM AR.SCRATCH.FILE
AR.INDEX.FIELD.LIST T)
(if CONDITION
then (AR.PROMPT.PRINT FORMWINDOW T "Can't get AR info
for AR # " AR.NUM)
(SETFILEPTR AR.SCRATCH.FILE START)
; Reset scratch pointer in case we started loading it
'DELETE
else (AR.PROMPT.PRINT FORMWINDOW T "analyzed AR # " AR.NUM)
INDEX.INFO))]
[for X in AR.NUM.DATA do
(RPLACA (CDR X)
(AR.INDEX.FIND.ENTRY.PTR (CAR X)
AR.NUM.DATA)])
```

(AR.INDEX.CREATE

[LAMBDA (FILENAME FIELD.LIST FORM.SPECS) ; Edited 21-Jul-88 14:36 by bvm

:: Create an empty AR index file.

```
(OR FIELD.LIST (SETQ FIELD.LIST AR.INDEX.DEFAULT.FIELDS))
(OR FORM.SPECS (SETQ FORM.SPECS AR.FORM.SPECS))
(LET ([FILE (OPENSTREAM FILENAME 'OUTPUT 'NEW '((TYPE BINARY)
(INDEX.DATA (create AR.INDEX.DATA
AR.INDEX.FILE _ NIL
AR.INDEX.ENTRY.BEGIN.PTR _ 0
AR.INDEX.ENTRY.END.PTR _ 0
AR.INDEX.FIELD.LIST _ FIELD.LIST))
(FIELD.SPECS (for X in FIELD.LIST collect (LIST X 'FIELD.BEGIN.PTR 0 'FIELD.END.PTR 0)))
(FIELD.PTR.OFFSET 4))
(for FIELD in FIELD.LIST bind ENUMERATED.FIELD.KEYS
do (if (SETQ ENUMERATED.FIELD.KEYS (AR.GET.ENUMERATED.FIELD.KEYS FORM.SPECS FIELD))
then (ARSPECPUT FIELD.SPECS FIELD 'ENUMERATED.FIELD.KEYLIST
(for FIELD.KEY in ENUMERATED.FIELD.KEYS as NUM from 1 join (LIST FIELD.KEY NUM)))
else (ARSPECPUT FIELD.SPECS FIELD 'FIELD.OFFSET FIELD.PTR.OFFSET)
(add FIELD.PTR.OFFSET 4)))
(replace (AR.INDEX.DATA AR.INDEX.FIELD.SPECS) of INDEX.DATA with FIELD.SPECS)
(replace (AR.INDEX.DATA AR.INDEX.ENTRY.SIZE) of INDEX.DATA with FIELD.PTR.OFFSET)
(SETFILEPTR FILE 0)
```

(PRINT INDEX.DATA FILE FILERDTBL) ; set DIR.FORMAT.PTR to 0
(\DWOUT FILE 0)
(CLOSEF FILE])

(AR.GET.ENUMERATED.FIELD.KEYS

[LAMBDA (FORM.SPECS FIELD) ; Edited 14-Feb-88 00:10 by bvm

;; Return all the valid keys for this field
(LET [(FIELD.SPEC (CDR (ASSOC FIELD FORM.SPECS)
(SELECTQ (LISTGET FIELD.SPEC 'FIELDTYPE)
(MENU (LISTGET FIELD.SPEC 'MENULIST))
(SUBMENU (CL:REMOVE-DUPLICATES (for X in (CDR (LISTGET FIELD.SPEC 'SUBMENULIST))
by (CDDR X) join (APPEND X))))))
NIL])

(AR.INDEX.FIND.ENTRY.PTR

[LAMBDA (NUM LOW.HINT HIGH.HINT) (* edited%: "21-Aug-84 14:37")

(PROG ((LOW (if LOW.HINT
else AR.INDEX.ENTRY.BEGIN.PTR))
(HIGH (if HIGH.HINT
else AR.INDEX.ENTRY.END.PTR))
LOW.NUM HIGH.NUM TEST.TEST.NUM)
(SETQ LOW.NUM (AR.GET.ENTRY.NUM LOW))
(SETQ HIGH.NUM (AR.GET.ENTRY.NUM HIGH))
(if (IGREATERP NUM HIGH.NUM)
then (SHOULDNT "Entry pointer higher than higher bound")))
loop
(if (EQ NUM LOW.NUM)
then (RETURN (CONS LOW T)))
(if (EQ NUM HIGH.NUM)
then (RETURN (CONS HIGH T)))
(SETQ TEST (IPLUS LOW (ITIMES (IQUOTIENT (IQUOTIENT (IDIFFERENCE HIGH LOW)
2)
AR.INDEX.ENTRY.SIZE)
AR.INDEX.ENTRY.SIZE)))
(if (EQ TEST LOW)
then (RETURN (CONS HIGH NIL)))
(SETQ TEST.NUM (AR.GET.ENTRY.NUM TEST))
(if (IGEQU NUM TEST.NUM)
then (SETQ LOW TEST)
(SETQ LOW.NUM TEST.NUM)
else (SETQ HIGH TEST)
(SETQ HIGH.NUM TEST.NUM))
(GO loop])

(AR.INDEX.REWRITE.ENTRY.DATA

[LAMBDA (NEW.FILE NUM.DATA.LIST) (* edited%: "16-Jul-84 15:55")

(PROG ((ENTRY.PTR AR.INDEX.ENTRY.BEGIN.PTR)
(FIELDS.WITH.OFFSETS (for FIELD.NAME in AR.INDEX.FIELD.LIST when (ARSPECGET AR.INDEX.FIELD.SPECS
FIELD.NAME 'FIELD.OFFSET)
collect FIELD.NAME))
FIELD.INCREMENT.LIST)
(SETQ FIELD.INCREMENT.LIST (for X in FIELDS.WITH.OFFSETS collect 0))
(until (AND (NULL NUM.DATA.LIST)
(IGEQU ENTRY.PTR AR.INDEX.ENTRY.END.PTR))
bind NUM.DATA NEXT.HIGHER.ENTRY.PTR REPLACE.FLG
do (SETQ NUM.DATA (CAR NUM.DATA.LIST))
(SETQ NEXT.HIGHER.ENTRY.PTR (CAR (CADR NUM.DATA)))
(SETQ REPLACE.FLG (CDR (CADR NUM.DATA)))
(if (OR (NULL NUM.DATA.LIST)
(IGREATERP NEXT.HIGHER.ENTRY.PTR ENTRY.PTR))
then ;; copy an existing AR entry, rather than create a new one
(SETFILEPTR AR.INDEX.FILE ENTRY.PTR) ; copy AR number to new entry
(\DWOUT NEW.FILE (\DWIN AR.INDEX.FILE))
; copy ptrs to various fields, adding on current increments
[for X in FIELD.INCREMENT.LIST do (\DWOUT NEW.FILE (IPLUS X (\DWIN AR.INDEX.FILE)
(SETQ ENTRY.PTR (GETFILEPTR AR.INDEX.FILE))
else ;; add a new AR entry from NUM.DATA.LIST
[if (NOT (EQ (CDDR NUM.DATA)
'DELETE))
then ; put out new number
(\DWOUT NEW.FILE (CAR NUM.DATA)) ; put out field ptrs for next higher field
[for FIELD.NAME in FIELDS.WITH.OFFSETS as X in FIELD.INCREMENT.LIST as FIELD.OFFSET
from 4 by 4 bind FIELD.BEGIN.PTR
do (SETQ FIELD.BEGIN.PTR (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME
'FIELD.BEGIN.PTR))
(\DWOUT NEW.FILE (IPLUS X (IDIFFERENCE (AR.GET.FIELD.VAL.PTR
NEXT.HIGHER.ENTRY.PTR
FIELD.NAME FIELD.OFFSET
FIELD.BEGIN.PTR)
FIELD.BEGIN.PTR]
; now, add field lengths to FIELD.INCREMENT.LIST

```

(for FIELD.NAME in FIELDS.WITH.OFFSETS as INC.LIST on FIELD.INCREMENT.LIST
 bind AR.FIELD.DATA do (SETQ AR.FIELD.DATA (ASSOC FIELD.NAME (CDDR NUM.DATA)))
 (if AR.FIELD.DATA
 then (RPLACA INC.LIST (IPLUS (CAR INC.LIST)
 (CADDR AR.FIELD.DATA))
 ;; if we are replacing an old AR, we must SUBTRACT the field lengths of the old AR from FIELD.INCREMENT.LIST
 (if REPLACE.FLG
 then (for INC.LIST on FIELD.INCREMENT.LIST as LENGTH.TO.BE.DELETED
 in (for FIELD.NAME in FIELDS.WITH.OFFSETS collect (AR.GET.FIELD.VAL.LENGTH
 NEXT.HIGHER.ENTRY.PTR
 FIELD.NAME))
 do (RPLACA INC.LIST (IDIFFERENCE (CAR INC.LIST)
 LENGTH.TO.BE.DELETED)))
 (SETQ ENTRY.PTR (IPLUS ENTRY.PTR AR.INDEX.ENTRY.SIZE)))
 (SETQ NUM.DATA.LIST (CDR NUM.DATA.LIST]))

```

(AR.INDEX.REWRITE.FIELD.DATA

```

[LAMBDA (NEWFILE SCRATCHFILE FIELD.NAME NUM.DATA.LIST) ; Edited 21-Jul-88 15:04 by bvm
 (PROG ((FIELD.KEYLIST (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'ENUMERATED.FIELD.KEYLIST))
 (FIELD.OFFSET (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'FIELD.OFFSET))
 (FIELD.DATA.BEGIN.PTR (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'FIELD.BEGIN.PTR))
 (FIELD.DATA.END.PTR (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'FIELD.END.PTR))
 DATA.PTR)
 (if (NOT (OR FIELD.KEYLIST FIELD.OFFSET))
 then (ERROR "Field doesn't have keylist or offset" FIELD.NAME))
 (SETQ DATA.PTR FIELD.DATA.BEGIN.PTR)
 (for NUM.DATA in NUM.DATA.LIST bind NEXT.HIGHER.ENTRY.PTR REPLACE.FLG NEXT.HIGHER.FIELD.VAL.PTR
 NUM.DATA.FOR.FIELD SCRATCH.FIELD.LEN
 do (SETQ NEXT.HIGHER.ENTRY.PTR (CAR (CADR NUM.DATA)))
 (SETQ REPLACE.FLG (CDR (CADR NUM.DATA)))
 (SETQ NEXT.HIGHER.FIELD.VAL.PTR (if FIELD.OFFSET
 then (AR.GET.FIELD.VAL.PTR NEXT.HIGHER.ENTRY.PTR FIELD.NAME
 FIELD.OFFSET FIELD.DATA.BEGIN.PTR
 FIELD.DATA.END.PTR)
 else (AR.ENTRY.PTR.TO.KEY.VAL.PTR NEXT.HIGHER.ENTRY.PTR
 FIELD.DATA.BEGIN.PTR)))
 (if (< DATA.PTR NEXT.HIGHER.FIELD.VAL.PTR)
 then (COPYBYTES AR.INDEX.FILE NEWFILE DATA.PTR NEXT.HIGHER.FIELD.VAL.PTR))
 (if (NOT (EQ (CDDR NUM.DATA)
 'DELETE))
 then (SETQ NUM.DATA.FOR.FIELD (ASSOC FIELD.NAME (CDDR NUM.DATA)))
 ;(field start length)
 (if NUM.DATA.FOR.FIELD
 then (SETQ SCRATCH.FIELD.LEN (CADDR NUM.DATA.FOR.FIELD))
 (SETFILEPTR SCRATCHFILE (CADR NUM.DATA.FOR.FIELD))
 (if FIELD.OFFSET
 then ; String field
 (if (> SCRATCH.FIELD.LEN 0)
 then (COPYBYTES SCRATCHFILE NEWFILE SCRATCH.FIELD.LEN))
 else ; Enumerated field
 (BOUT NEWFILE (OR (LISTGET FIELD.KEYLIST (PACKC (AR.READ.BYTES
 SCRATCHFILE
 SCRATCH.FIELD.LEN)
 0))))
 ; Empty enumerated field--all must be present (string fields can
 ; be sparse).
 (BOUT NEWFILE 0)))
 (SETQ DATA.PTR (if REPLACE.FLG
 then (if FIELD.OFFSET
 then (AR.GET.FIELD.VAL.PTR (+ NEXT.HIGHER.ENTRY.PTR
 AR.INDEX.ENTRY.SIZE)
 FIELD.NAME FIELD.OFFSET FIELD.DATA.BEGIN.PTR
 FIELD.DATA.END.PTR)
 else (ADD1 NEXT.HIGHER.FIELD.VAL.PTR))
 else NEXT.HIGHER.FIELD.VAL.PTR))
 (if (< DATA.PTR FIELD.DATA.END.PTR)
 then (COPYBYTES AR.INDEX.FILE NEWFILE DATA.PTR FIELD.DATA.END.PTR))

```

(AR.QFORM.FN.PRINT.INDEX

```

[LAMBDA (QFORMWINDOW) ; Edited 16-Feb-88 22:36 by bvm
 (WITH.AR.QUERY QFORMWINDOW (TTY.PROCESS (THIS.PROCESS))
 (AR.INDEX.PRINT T)
 (AR.PROMPT "done" QFORMWINDOW))

```

(AR.INDEX.PRINT

```

[LAMBDA (FILE PRINT.ENTRY.DATA.FLG) ; Edited 15-Feb-88 18:37 by bvm
 (LET
 ((*PRINT-BASE* 10))
 (printout FILE "Total file size: " (GETEOPTR AR.INDEX.FILE)
 " bytes" T T)
 (printout FILE "Total Field Space: " .TAB 20 AR.INDEX.ENTRY.BEGIN.PTR " bytes" T)

```

```

(for FIELD.NAME in AR.INDEX.FIELD.LIST bind FIELD.BYTES do [SETQ FIELD.BYTES (- (ARSPECGET
AR.INDEX.FIELD.SPECS
FIELD.NAME
'FIELD.END.PTR)
(ARSPECGET
AR.INDEX.FIELD.SPECS
FIELD.NAME
'FIELD.BEGIN.PTR]
(printout FILE FIELD.NAME .TAB 20 FIELD.BYTES T))
(printout FILE T "Total Entry Space: " (- AR.INDEX.ENTRY.END.PTR AR.INDEX.ENTRY.BEGIN.PTR)
" bytes" T)
(printout T (IQUOTIENT (- AR.INDEX.ENTRY.END.PTR AR.INDEX.ENTRY.BEGIN.PTR)
AR.INDEX.ENTRY.SIZE)
" entries of " AR.INDEX.ENTRY.SIZE " bytes" T)
(if (EQ PRINT.ENTRY.DATA.FLG 'ALL)
then [for ENTRY.PTR from AR.INDEX.ENTRY.BEGIN.PTR by AR.INDEX.ENTRY.SIZE until (>= ENTRY.PTR
AR.INDEX.ENTRY.END.PTR)
do (printout FILE "Entry # " (PROGN (SETFILEPTR AR.INDEX.FILE ENTRY.PTR)
(\DWIN AR.INDEX.FILE))
T)
(for FIELD.NAME in AR.INDEX.FIELD.LIST bind VAL.DATA FIELD.KEYLIST VAL.NUM
do (LET* [(FIELD.SPEC (CDR (ASSOC FIELD.NAME AR.INDEX.FIELD.SPECS)))
(FIELD.BEGIN.PTR (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR))
(FIELD.OFFSET (LISTGET FIELD.SPEC 'FIELD.OFFSET)
(if FIELD.OFFSET
then (DESTRUCTURING-BIND (PTR . LEN)
(AR.GET.FIELD.VAL.SHAPE ENTRY.PTR FIELD.OFFSET FIELD.BEGIN.PTR
(LISTGET FIELD.SPEC 'FIELD.END.PTR))
(printout FILE FIELD.NAME " %")
(SETFILEPTR AR.INDEX.FILE PTR)
(COPYBYTES AR.INDEX.FILE FILE LEN)
(printout FILE "% " T))
else (SETQ FIELD.KEYLIST (LISTGET FIELD.SPEC 'ENUMERATED.FIELD.KEYLIST))
(SETFILEPTR AR.INDEX.FILE (AR.ENTRY.PTR.TO.KEY.VAL.PTR ENTRY.PTR
FIELD.BEGIN.PTR))
(printout FILE FIELD.NAME " %")
(SETQ VAL.NUM (BIN AR.INDEX.FILE))
(if (NEQ VAL.NUM 0)
then (printout FILE (for X on FIELD.KEYLIST by (CDDR X)
when (EQ VAL.NUM (CADR X))
do (RETURN (CAR X))
(printout FILE "% " T)
elseif PRINT.ENTRY.DATA.FLG
then (printout FILE "Contains entries: ")
(for ENTRY.PTR from AR.INDEX.ENTRY.BEGIN.PTR by AR.INDEX.ENTRY.SIZE until (>= ENTRY.PTR
AR.INDEX.ENTRY.END.PTR)
do (printout FILE (PROGN (SETFILEPTR AR.INDEX.FILE ENTRY.PTR)
(\DWIN AR.INDEX.FILE))
% ,))
(TERPRI FILE)])

```

(AR.QFORM.FN.UPDATE

```

[LAMBDA (QFORMWINDOW) (* mjs "8-Aug-84 15:18")
(PROG ((ULIST (AR.GET.BUTTON.FIELD.AS.LIST QFORMWINDOW 'Update List:))
VAL)
(SETQ VAL (AR.INDEX.UPDATE QFORMWINDOW ULIST))
(AR.PROMPT (LIST "Update done --- new file: " VAL)
QFORMWINDOW])

```

(AR.INDEX.UPDATE

```

[LAMBDA (FORMWINDOW AR.NUM.LIST) ; Edited 21-Jul-88 15:07 by bvm
;; Update the AR index with changed ars listed in AR.NUM.LIST
(WITH.AR.QUERY FORMWINDOW (PROG (*UPPER-CASE-FILE-NAMES* AR.NUM.DATA AR.SCRATCH.FILE NEW.AR.INDEX.FILE
NEW.AR.INDEX.DATA NEW.FIELD.SPECS NEW.AR.INDEX.DATA.PTR)
(if [NOT (AND (LISTP AR.NUM.LIST)
(EVERY AR.NUM.LIST (FUNCTION FIXP)
then (AR.PROMPT.PRINT FORMWINDOW T "Bad ar number list")
(RETURN))
(SETQ AR.NUM.LIST (SORT (CL:REMOVE-DUPLICATES AR.NUM.LIST)))
[SETQ AR.SCRATCH.FILE (OPENSTREAM (PACKFILENAME.STRING 'VERSION NIL
'BODY
'AR.TEMP
'BODY
(FULLNAME AR.INDEX.FILE))
' BOTH
' NEW
' ((TYPE BINARY]
(SETQ AR.NUM.DATA (AR.GATHER.NEW.AR.DATA FORMWINDOW AR.NUM.LIST
AR.SCRATCH.FILE))
; Read the changed ar's data
[SETQ NEW.AR.INDEX.FILE (OPENSTREAM (PACKFILENAME.STRING 'VERSION NIL
'BODY
'ARINDEX.NEW
'BODY

```

```

(FULLNAME AR.INDEX.FILE))
      ' OUTPUT
      ' NEW
      ' ((TYPE BINARY]
; Create a new index file
(SETQ NEW.AR.INDEX.DATA
  (create AR.INDEX.DATA
    AR.INDEX.FILE _ NIL
    AR.INDEX.FIELD.LIST _ AR.INDEX.FIELD.LIST
    AR.INDEX.ENTRY.SIZE _ AR.INDEX.ENTRY.SIZE))
(SETQ NEW.FIELD.SPECS (COPYALL AR.INDEX.FIELD.SPECS))
(for FIELD.NAME in AR.INDEX.FIELD.LIST
  do (ARSPECPUT NEW.FIELD.SPECS FIELD.NAME 'FIELD.BEGIN.PTR (GETFILEPTR
                                                    NEW.AR.INDEX.FILE
                                                    ))
    (AR.INDEX.REWRITE.FIELD.DATA NEW.AR.INDEX.FILE AR.SCRATCH.FILE
      FIELD.NAME AR.NUM.DATA)
    ; Write new or copy old data
    (ARSPECPUT NEW.FIELD.SPECS FIELD.NAME 'FIELD.END.PTR (GETFILEPTR
                                                          NEW.AR.INDEX.FILE
                                                          )))
(DELFILE (CLOSEF AR.SCRATCH.FILE))
(replace (AR.INDEX.DATA AR.INDEX.FIELD.SPECS) of NEW.AR.INDEX.DATA
  with NEW.FIELD.SPECS)
(replace (AR.INDEX.DATA AR.INDEX.ENTRY.BEGIN.PTR) of NEW.AR.INDEX.DATA
  with (GETFILEPTR NEW.AR.INDEX.FILE))
(AR.INDEX.REWRITE.ENTRY.DATA NEW.AR.INDEX.FILE AR.NUM.DATA)
(replace (AR.INDEX.DATA AR.INDEX.ENTRY.END.PTR) of NEW.AR.INDEX.DATA
  with (GETFILEPTR NEW.AR.INDEX.FILE))
(SETQ NEW.AR.INDEX.DATA.PTR (GETFILEPTR NEW.AR.INDEX.FILE))
(PRINT NEW.AR.INDEX.DATA NEW.AR.INDEX.FILE FILERDTBL)
(\DWOUT NEW.AR.INDEX.FILE NEW.AR.INDEX.DATA.PTR)
(CLOSEF NEW.AR.INDEX.FILE)
(RETURN (RENAMEFILE (FULLNAME NEW.AR.INDEX.FILE)
  (PACKFILENAME.STRING 'VERSION NIL 'BODY (FULLNAME
    AR.INDEX.FILE]
)
)
)

```

(DEFINEQ

(AR.GET.FIELD.VAL.LENGTH

```

[LAMBDA (ENTRY.PTR FIELD.NAME FIELD.OFFSET FIELD.VAL.BEGIN.PTR FIELD.VAL.END.PTR)
  (* edited%: "13-Jul-84 14:45")
  (if (ILESSP ENTRY.PTR AR.INDEX.ENTRY.END.PTR)
    then (PROG ((NEXT.ENTRY.PTR (IPLUS ENTRY.PTR AR.INDEX.ENTRY.SIZE))
      CURRENT.RELPTR NEXT.RELPTR)
      [if (NULL FIELD.OFFSET)
        then (SETQ FIELD.OFFSET (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'FIELD.OFFSET))
      (SETFILEPTR AR.INDEX.FILE (IPLUS ENTRY.PTR FIELD.OFFSET))
      (SETQ CURRENT.RELPTR (\DWIN AR.INDEX.FILE))
      [SETQ NEXT.RELPTR (if (ILESSP NEXT.ENTRY.PTR AR.INDEX.ENTRY.END.PTR)
        then (SETFILEPTR AR.INDEX.FILE (IPLUS NEXT.ENTRY.PTR FIELD.OFFSET))
        (\DWIN AR.INDEX.FILE)
        else (IDIFFERENCE (if FIELD.VAL.END.PTR
          else (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME
            'FIELD.END.PTR))
        (if FIELD.VAL.BEGIN.PTR
          else (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME
            'FIELD.BEGIN.PTR]
      (RETURN (IDIFFERENCE NEXT.RELPTR CURRENT.RELPTR)))
    else 0])

```

(AR.GET.FIELD.VAL.PTR

```

[LAMBDA (ENTRY.PTR FIELD.NAME FIELD.OFFSET FIELD.VAL.BEGIN.PTR FIELD.VAL.END.PTR)
  (* edited%: "13-Jul-84 15:41")
  (if (ILESSP ENTRY.PTR AR.INDEX.ENTRY.END.PTR)
    then [SETFILEPTR AR.INDEX.FILE (IPLUS ENTRY.PTR
      else (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME
        'FIELD.OFFSET])
      (IPLUS (if FIELD.VAL.BEGIN.PTR
        else (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'FIELD.BEGIN.PTR))
      (\DWIN AR.INDEX.FILE))
    else (if FIELD.VAL.END.PTR
      else (ARSPECGET AR.INDEX.FIELD.SPECS FIELD.NAME 'FIELD.END.PTR])

```

(AR.GET.FIELD.VAL.SHAPE

```

[LAMBDA (ENTRY.PTR FIELD.OFFSET FIELD.BEGIN FIELD.END) ; Edited 15-Feb-88 18:36 by bvm
;; Returns a pair (filepointer . length) describing the location and size of the text field described by the args.
(if (< ENTRY.PTR AR.INDEX.ENTRY.END.PTR)
  then ;; Good entry value. The text strings for this field are all stored contiguously in the region between FIELD.VAL.BEGIN.PTR and

```

;; FIELD.VAL.END.PTR. The index entry pointed to by ENTRY.PTR contains offsets within that region.

```
(LET ((PTRLOC (+ ENTRY.PTR FIELD.OFFSET))
      THISPTR)
      (SETFILEPTR AR.INDEX.FILE PTRLOC)
      (SETQ THISPTR (\DWIN AR.INDEX.FILE)) ; Offset in the region for this field for this ar.
      (CONS (+ THISPTR FIELD.BEGIN)
            (- (if (<= (- AR.INDEX.ENTRY.END.PTR PTRLOC)
                      AR.INDEX.ENTRY.SIZE)
                then ; Fetching info for last ar in index, so the field goes until the end
                    (- FIELD.END FIELD.BEGIN) ; of this text region
                else ; Get start of NEXT ar's info
                    (SETFILEPTR AR.INDEX.FILE (+ PTRLOC AR.INDEX.ENTRY.SIZE))
                    (\DWIN AR.INDEX.FILE)
                    THISPTR)))
      ; Not an ar. This is for continuity, I guess
      (CONS FIELD.END 0]))
```

(AR.GET.ENTRY.NUM

```
[LAMBDA (PTR) ; (* edited%: "13-Jul-84 11:42")
  (if (IGEQL PTR AR.INDEX.ENTRY.END.PTR)
      then MAX.FIXP
      else (SETFILEPTR AR.INDEX.FILE PTR)
           (\DWIN AR.INDEX.FILE]))
)
```

(RPAQ? AR.INDEX.DEFAULT.FIELDS ' (Subject%: Source%: Date%: Submitter%: |Assigned To:| Attn%: Status%: In/By%:
|Problem Type:| Impact%: Difficulty%: Frequency%: Priority%: System%:
Subsystem%: Machine%: Disk%: |Lisp Version:| |Source Files:|
|Microcode Version:| |Memory Size:| |File Server:|
|Server Software Version:| Edit-By%: Edit-Date%:))

(DECLARE%: EVAL@COMPILE DONTCOPY

(FILESLOAD (LOADCOMP)
ARQUERY)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS AR.INDEX.DEFAULT.FIELDS)
)

(DEFMACRO AR.ENTRY.PTR.TO.KEY.VAL.PTR (ENTRYPTR BEGINPTR)
`(+ (IQUOTIENT (- ,ENTRYPTR AR.INDEX.ENTRY.BEGIN.PTR)
AR.INDEX.ENTRY.SIZE)
,BEGINPTR))

(DEFMACRO ARSPECPUT (SPECS FIELDNAME PROP NEWVALUE)
` (LISTPUT (CDR (ASSOC ,FIELDNAME ,SPECS))
,PROP
,NEWVALUE))
)

(PUTPROPS ARINDEX COPYRIGHT ("Venue & Xerox Corporation" 1988 1990 1992))

FUNCTION INDEX

AR.GATHER.NEW.AR.DATA	1	AR.GET.FIELD.VAL.SHAPE	5	AR.INDEX.REWRITE.FIELD.DATA	3
AR.GET.ENTRY.NUM	6	AR.INDEX.CREATE	1	AR.INDEX.UPDATE	4
AR.GET.ENUMERATED.FIELD.KEYS	2	AR.INDEX.FIND.ENTRY.PTR	2	AR.QFORM.FN.PRINT.INDEX	3
AR.GET.FIELD.VAL.LENGTH	5	AR.INDEX.PRINT	3	AR.QFORM.FN.UPDATE	4
AR.GET.FIELD.VAL.PTR	5	AR.INDEX.REWRITE.ENTRY.DATA	2		

MACRO INDEX

AR.ENTRY.PTR.TO.KEY.VAL.PTR	6	ARSPECPUT	6
-----------------------------------	---	-----------------	---

VARIABLE INDEX

AR.INDEX.DEFAULT.FIELDS	6
-------------------------------	---
