

File created: 28-Sep-90 15:09:52 {DSK}<usr>local>lde>lispcore>internal>library>ARQUERY.;2

changes to: (VARS ARQUERYCOMS)

previous date: 15-Jun-90 11:31:57 {DSK}<usr>local>lde>lispcore>internal>library>ARQUERY.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

::
:: Copyright (c) 1988, 1989, 1990 by Venue & Xerox Corporation. All rights reserved.

(RPAQQ **ARQUERYCOMS**

```
[ (COMS
                                ; Query window management
(FNS AR.QFORM.CREATE AR.QFORM.GROUP.CREATE AR.QFORM.GET.DEFAULT.INDEX AR.QFORM.CREATE.ABORT
AR.QFORM.GDATE AR.QUERY.WHENSELECTEDFN AR.QUERY.CLOSEFN AR.QUERY.SHRINKFN
AR.QUERY.CLOSE/SHRINK AR.QUERY.EXPANDFN AR.QFORM.ICONFN AR.INDEX.OPEN AR.INDEX.FILE.REOPEN
AR.INDEX.FILE.CLOSE AR.QFORM.QUERY AR.QFORM.BUTTONFN AR.GET.QLIST.PROMPT.MENU
AR.QLIST.MENU.COMPARISONS AR.QFORM.PROMPT.LIST.FN AR.QFORM.TITLEMENU
AR.MAKE.COMPARISON.STRING AR.GET.BUTTON.FIELD.AS.LIST))
(COMS
                                ; AR Browser window stuff
(FNS AR.BROWSER.PRINTFN AR#.FROM.ITEM AR.BROWSER.COMMANDFN AR.BROWSER.DO.COMMAND
AR.BROWSER.SELECTED.ARS AR.BROWSER.DISPLAY AR.BROWSER.EDIT AR.BROWSER.HARDCOPY))
(COMS
                                ; Sorting
(FNS AR.QFORM.SORT AR.SORT.BY AR.GET.SLIST.PROMPT.MENU AR.ENSURE.QUERY.FIELDS AR.ENSURE.QUERY.DATA
AR.COLLECT.ENTRY.FIELDS AR.ENSURE.QUERY.DATA.ITEM AR.AUGMENT.QUERY.FIELDS
AR.KEYVALS.FROM.KEYLIST))
(COMS
                                ; Printing summaries
(FNS AR.QFORM.SUMMARY AR.QFORM.SUMMARY.TEXT AR.MAKE.SUMMARY.FILE AR.MAKE.SUMMARY.TEXT.FILE
AR.QFORM.SUMMARY.TEDIT AR.QFORM.SUMMARIZE.CHECK AR.OPEN.IP.STREAM AR.PRINT.PADDED
AR.IP.FROM.SUMMARY)
(FNS AR.PRINT.SUMMARY AR.PRINT.SUMMARY.FIELD))
(COMS
                                ; Evaluating AR queries
(FNS AR.QUERY.AR.QUERY.SMALLP AR.QUERY.EVAL AR.BAD.QUERY AR.QUERY.AND AR.QUERY.NAND
AR.QUERY.SORT.CLAUSES AR.QUERY.SORT.ORDER AR.QUERY.SORT.VALUE AR.QUERY.OR
AR.QUERY.COMBINE.RESULT)
(FNS AR.QUERY.IS AR.QUERY.IS.EXACTLY AR.QUERY.COMPARE.ENUMERATED AR.QUERY.IS.EMPTY)
(FNS AR.QUERY.HAS AR.COLLECT.SHAPES AR.COLLECT.SIZES AR.SPARSE.QUERY AR.INDICES.FROM.FILEPTRS)
(FNS AR.QUERY.COMPARE AR.QUERY.COMPARE.PARSE AR.QUERY.NUMBER AR.QUERY.PRODUCE.INDEXES AR.COLLECT.N
AR.INDEX.FROM.NUMBER)
(FNS AR.QUERY.DATE AR.QUERY.GENERAL.DATE AR.QUERY.PARSE.DATES AR.INDEX.FROM.DATE
AR.DATE.FROM.INDEX)
(FNS AR.NUMS.FROM.QUERY AR.ENTRY.PTR.FROM.INDEX AR.ENTRY.VALUE.FROM.INDEX AR.ENTRY.VALUE.NEXT
AR.SELECT.WINDOW))
[COMS
                                ; Patch for nasty bug in \INCFILEPTR
(FNS AR.INCFILEPTR)
(DECLARE%: DONTEVAL@LOAD DOCOPY (P (AND (CCODEP 'AR.INCFILEPTR (MOVD 'AR.INCFILEPTR
' \PAGED.INCFILEPTR NIL T)
(COMS
;; Set up file names. We use VARS on AR.INDEX.DEFAULT.FILE.NAME to force it correct in the case where the index is moving. If
;; user has set it to some disk file for manual caching, make that the cache name
(INITVARS (AR.INDEX.CACHE.FILE.NAME (AND (BOUNDP 'AR.INDEX.DEFAULT.FILE.NAME)
(STRPOS "DSK" (UNPACKFILENAME.STRING
AR.INDEX.DEFAULT.FILE.NAME
'HOST)
NIL NIL T NIL UPPERARRAY)
AR.INDEX.DEFAULT.FILE.NAME))
(AR.ALWAYS.CACHE.INDEX :ASK))
(VARS (AR.INDEX.DEFAULT.FILE.NAME "{AR:MV:Envos}<LispARs>AR.INDEX"))
(VARS (AR.QFORM.TITLEMENU)
AR.QFORM.FORMAT AR.QFORM.SPECS AR.QFORM.ICON AR.COMPARISON.OPERATORS)
[INITVARS [AR.BROWSER.MENU.ITEMS '(("Display" AR.BROWSER.DISPLAY "Display selected AR in a readonly
window")
("Edit" AR.BROWSER.EDIT "Edit selected AR in an ARedit window
(uses same window as last time unless you select with middle
button).")
("Hardcopy AR(s)" AR.BROWSER.HARDCOPY "Make hardcopy of the complete
content of the selected AR(s)"]
[AR.QUERY.MENU.ITEMS '(("Query" (AR.QFORM.QUERY)
"Search the AR database for ARs matching the Query List")
("Sort" AR.QFORM.SORT "Sort the ARs in the browser window using the new
Sort List")
("Hardcopy Summary" AR.QFORM.SUMMARY "Print to your default printer a
summary of the ARs displayed in the browser"
(SUBITEMS ("Text Summary" AR.QFORM.SUMMARY.TEXT "Make a plain text
version of the summary on a file")
("Tedit Summary" AR.QFORM.SUMMARY.TEDIT "Edit (using Tedit)
a plain text version of the summary"]
(AR.WHENSELECTEDSHADE 4672)
[AR.DISPLAY.FIELDS '((Status%: 5)
(Subject%: 50)
(Attn%: 15)
(System%: 13)
```

```

      (Subsystem%: 13]
[AR.SUMMARY.FIELDS '( (Date%: 9 T)
                    (System%: 13 T)
                    (Subsystem%: 14)
                    (Status%: 10 T)
                    (Attn%: 13)
                    (Subject%: 55)
                    (Priority%: 10)
                    (Difficulty%: 10)
                    (Impact%: 8)
                    (|Problem Type:| 13])
(AR.TEDIT.FIELDS)
(AR.SUMMARY.MIN.LINES 2)
(AR.CLEANUP.SORT.ORDER '(System%: Subsystem%: Status%: Priority%: Impact%:))
(AR.SORT.EQUIVALENTS '((Status%: (Open Open/Unreleased)
(ADDVARS (AR.SORT.SPEC.ITEMS ("Standard Summary Order" [FUNCTION (LAMBDA NIL AR.CLEANUP.SORT.ORDER)
                          "Sort order used by AR Cleanup when producing personal summaries."))
(AR.QUERY.SPEC.ITEMS ("Status is UnFixed" "(OR (Status: >= Open/Unreleased) (Status: =
                          Incomplete))" "AR is somehow Open, i.e., not Fixed, Declined or
                          Obsoleted")
                          ("Status is Resolved" "(AND (Status: >= Obsolete) (Status: <= Fixed)" "AR has been taken
                          care of--Fixed, Declined, etc.")
                          ("Mandatory" "(AND (Status: >= Open/Unreleased) (Priority: = Absolutely) (Problem% Type:
                          ~= Feature)" "Non-Feature AR has priority Absolutely and is still open somehow"))
(DECLARE%: EVAL@COMPILE DONTCOPY (RECORDS AR.INDEX.DATA ARQUERYDATA ARINDEXDESCR)
(GLOBALVARS AR.QFORM.ICON AR.BROWSER.MENU.ITEMS AR.QUERY.MENU.ITEMS AR.COMPARISON.OPERATORS
AR.QFORM.TITLEMENU)
(LOCALVARS . T)
(FUNCTIONS WITH.AR.QUERY ARSPECGET)
[P [CL:PROCLAIM (CONS 'CL:SPECIAL (RECORDFIELDNAMES 'AR.INDEX.DATA)
(CASE DFNFLG
((PROP ALLPROP)
; When I load this file PROP, need to get these defs evaled,
; grumble

[LET ((DFNFLG T))
(MAPC '(WITH.AR.QUERY ARSPECGET)
(FUNCTION (LAMBDA (FN)
(CL: EVAL (GETDEF FN 'FUNCTIONS NIL '(NOERROR)))
; These aren't ours, but declare them to reduce the warnings
; from compiler & masterscope
(CL:PROCLAIM '(CL:SPECIAL DEFAULTFONT DEFAULTLANDPAGEREGION)
(CONSTANTS (AR.BYTES.PER.PTR 4))
(FILES (SOURCE)
TABLEBROWSERDECLS))
(DECLARE%: EVAL@COMPILE DONTCOPY
(P (CL:PROCLAIM '(CL:SPECIAL AR.INDEX.DEFAULT.FILE.NAME AR.INDEX.CACHE.FILE.NAME
AR.ALWAYS.CACHE.INDEX AR.QFORM.SPECS AR.QFORM.FORMAT AR.WHENSELECTEDSHADE
AR.DISPLAY.FIELDS AR.SUMMARY.MIN.LINES AR.SUMMARY.FIELDS AR.TEDIT.FIELDS
AR.QUERY.SPEC.ITEMS AR.SORT.SPEC.ITEMS AR.SORT.EQUIVALENTS])

```

:: Query window management

(DEFINEQ

(AR.QFORM.CREATE

```

[LAMBDA (AR.INDEX.FILE.NAME WINDOW DONTSPAWN) ; Edited 25-Feb-87 10:47 by jds
;; Create an AR query form. Queries will be done against AR.INDEX.FILE.NAME. WINDOW, if supplied, will be used as the main query window.
;; If DONTSPAWN is T, this'll be completed before the function returns; otherwise it'll be spawned as an asynchronous process.
(COND
(DONTSPAWN ; Want the window created before returning.
(AR.QFORM.GROUP.CREATE AR.INDEX.FILE.NAME WINDOW))
(T ; Let the caller go ahead, while we make the window on our own
; time.
(ADD.PROCESS (LIST (FUNCTION AR.QFORM.GROUP.CREATE)
(KWOTE AR.INDEX.FILE.NAME)
(KWOTE WINDOW))
' NAME
' AR.QUERY.FORM.TEMP])

```

(AR.QFORM.GROUP.CREATE

```

[LAMBDA (INDEX.FILENAME WINDOW NO.BROWSER) ; Edited 4-Aug-88 12:52 by bvm

```

;;; Set up a query-window group (main window, summary browser, and prompt window). Queries will be done against AR.INDEX.FILE.NAME.
 ;;; WINDOW, if supplied, will be used as the query window. If NO.BROWSER is true, this window is being created solely to hang queries off, so only the
 ;;; main window and a prompt window are supplied.

```

(LET* ([BROWSERMENUW (MENUWINDOW (create MENU
ITEMS _ AR.BROWSER.MENU.ITEMS
MENUFONT _ ARBOLDFONT
CENTERFLG _ T
MENURROWS _ 1
WHENSELECTEDFN _ (FUNCTION AR.BROWSER.COMMANDFN)
[QUERYMENUW (MENUWINDOW (create MENU
ITEMS _ AR.QUERY.MENU.ITEMS
MENUFONT _ ARBOLDFONT

```

```

                CENTERFLG _ T
                MENUROWS _ 1
                WHENSELECTEDFN _ (FUNCTION AR.BROWSER.COMMANDFN]
(MENUHEIGHT (WINDOWPROP QUERYMENUW 'HEIGHT))
(FONTHEIGHT (FONTPROP (OR (WINDOWP WINDOW)
                DEFAULTFONT)
                'HEIGHT))
(PROMPHEIGHT (HEIGHTIFWINDOW (TIMES 2 FONTHEIGHT)))
(BROWSERHEIGHT (HEIGHTIFWINDOW (TIMES 8 FONTHEIGHT)
                T))
(QUERYHEIGHT (HEIGHTIFWINDOW [TIMES 3 (+ 2 (FONTPROP ARBOLDFONT 'HEIGHT)
                T))
QFORMWINDOW QREG REG QFORM.ENTRY.WINDOW DATA)
;; set up main window. I assume the two menus are the same height
(if (NOT (WINDOWP WINDOW))
    then (LET ((OTHERHEIGHTS (+ QUERYHEIGHT MENUHEIGHT MENUHEIGHT PROMPHEIGHT)))
        ; Height of all the fixed window parts
        [SETQ REG (OR (REGIONP WINDOW)
                    (GETREGION 400 (+ BROWSERHEIGHT OTHERHEIGHTS)
                    (SETQ QFORMWINDOW (CREATEW (SETQ QREG (create REGION using REG HEIGHT _ QUERYHEIGHT BOTTOM
                    _ (+ (fetch (REGION BOTTOM)
                    of REG)
                    BROWSERHEIGHT MENUHEIGHT))
                    )
                    (CONCAT AR.IDENTIFICATION.STRING " Query Specification"))
        (replace (REGION HEIGHT) of REG with (- (fetch (REGION HEIGHT) of REG)
        OTHERHEIGHTS)))
    else (SETQ QREG (WINDOWPROP (SETQ QFORMWINDOW WINDOW)
        'REGION))
        (SETQ REG (create REGION
            LEFT _ (fetch (REGION LEFT) of QREG)
            BOTTOM _ (- (fetch (REGION BOTTOM) of QREG)
            BROWSERHEIGHT MENUHEIGHT)
            WIDTH _ (fetch (REGION WIDTH) of QREG)
            HEIGHT _ BROWSERHEIGHT))
        (WINDOWPROP QFORMWINDOW 'AR.WINDOW.PROC.NAME 'AR.QUERY.FORM)
        (WINDOWPROP QFORMWINDOW 'MINSIZE (CONS 200 QUERYHEIGHT))
        (WINDOWPROP QFORMWINDOW 'MAXSIZE (CONS MAX.SMALLP (fetch (REGION HEIGHT) of QREG)))
        (WINDOWPROP QFORMWINDOW 'ICONFN (FUNCTION AR.QFORM.ICONFN))
        (WINDOWADDPROP QFORMWINDOW 'SHRINKFN (FUNCTION AR.QUERY.SHRINKFN)
            T)
        (WINDOWADDPROP QFORMWINDOW 'EXPANDFN (FUNCTION AR.QUERY.EXPANDFN)
            T)
;; Attach query operations menu
(ATTACHWINDOW QUERYMENUW QFORMWINDOW 'TOP 'JUSTIFY)
(GETPROMPTWINDOW QFORMWINDOW 2)
(if [SETQ DATA (AR.INDEX.OPEN QFORMWINDOW (OR INDEX.FILENAME (AR.QFORM.GET.DEFAULT.INDEX QFORMWINDOW)
    then (WINDOWPROP QFORMWINDOW 'AR.INDEX.DATA DATA)
        (WINDOWPROP QFORMWINDOW 'AR.INDEX.MONITORLOCK (CREATE.MONITORLOCK "AR Index"))
        [if (NOT NO.BROWSER)
            then
                ; Add browser window and its menu, and install query menu in
                ; query window
                (ATTACHWINDOW BROWSERMENUW QFORMWINDOW 'BOTTOM 'JUSTIFY)
                (SETQ QFORM.ENTRY.WINDOW (CREATEW REG (CONCAT AR.IDENTIFICATION.STRING " Query
                Browser")))
                (ATTACHWINDOW QFORM.ENTRY.WINDOW QFORMWINDOW 'BOTTOM 'JUSTIFY)
                ; Browser window goes on very bottom so that the scroll bar
                ; doesn't get in the way
                (WINDOWPROP QFORMWINDOW 'QFORM.ENTRY.WINDOW QFORM.ENTRY.WINDOW)
                (WINDOWPROP QFORM.ENTRY.WINDOW 'MINSIZE (CONS 10 (HEIGHTIFWINDOW (TIMES 2 FONTHEIGHT)
                T)))
                (AR.FORM.CREATE QFORMWINDOW ARBOLDFONT AR.QFORM.SPECS AR.QFORM.FORMAT
                (LIST 'TITLEMENUFN (FUNCTION AR.QFORM.TITLEMENU)
                ; Don't install CLOSEFN til now, so that we can override TEdit's
                (WINDOWADDPROP QFORMWINDOW 'CLOSEFN (FUNCTION AR.QUERY.CLOSEFN)
                T])

```

(AR.QFORM.GET.DEFAULT.INDEX

[LAMBDA (QFORMWINDOW) ; Edited 4-Aug-88 12:53 by bvm

;; Returns the file name of the index to open for QFORMWINDOW. This fusses about caching.

```

(if (OR (NULL AR.INDEX.CACHE.FILE.NAME)
    (NULL AR.ALWAYS.CACHE.INDEX))
    then
        ; No cache, or we're supposed to ignore it, go straight to the
        ; master
        AR.INDEX.DEFAULT.FILE.NAME
    else
        (LET
            ((*UPPER-CASE-FILE-NAMES* NIL))
            (WINDOWPROP QFORMWINDOW 'PROCESS (THIS.PROCESS))
            (WINDOWADDPROP QFORMWINDOW 'CLOSEFN (FUNCTION AR.QFORM.CREATE.ABORT))
            ; Arrange to go away if user aborts by closing window.
            (CL:UNWIND-PROTECT
                [PROG ((MASTER (INFILEP AR.INDEX.DEFAULT.FILE.NAME))

```

```

(CACHE (INFILEP AR.INDEX.CACHE.FILE.NAME))
MASTERDATE CACHEDATE CLOSEHACK)
(if (NULL MASTER)
  then (AR.PROMPT.PRINT QFORMWINDOW "Can't find " AR.INDEX.DEFAULT.FILE.NAME " so will use
        cache")
        (RETURN (OR CACHE AR.INDEX.CACHE.FILE.NAME))
  elseif (NULL CACHE)
  then (PRINTOUT QFORMWINDOW "Local cache " AR.INDEX.CACHE.FILE.NAME " does not yet exist")
  elseif (AND (SETQ CACHEDATE (GETFILEINFO CACHE 'ICREATIONDATE))
            (SETQ MASTERDATE (GETFILEINFO MASTER 'ICREATIONDATE))
            (>= CACHEDATE MASTERDATE))
  then ; Cache is up to date
        (RETURN CACHE)
  else (CL:FORMAT QFORMWINDOW "Local cache (~A) is older than master index (~A)" (
                                             AR.QFORM.GDATE
                                             CACHEDATE)
        (AR.QFORM.GDATE MASTERDATE)))

(RETURN
 (PROG1 (SELECTQ [COND
                ((EQ AR.ALWAYS.CACHE.INDEX T)
                 :COPY)
                (T
                 ; Ask user whether to cache
                 (LET* [(CHOICEMENU (create MENU
                                         ITEMS _
                                         `["Copy master index to local cache" :COPY
                                           "Copy the master index file to the
                                           local cache file (this will take a
                                           while), then use the local cache."]
                                         ("Use master index directly" :NEW "Use the
                                           master index directly, without
                                           caching it.")
                                         ,@(AND CACHE '("Use local cache (ignore
                                           master)" :OLD "Use the local
                                           index cache, even
                                           though there is a
                                           newer master index"]
                                         CENTERFLG _ T
                                         MENUFONT _ ARBOLDFONT
                                         MENUOUTLINESIZE _ 4)
                 (REG (WINDOWPROP QFORMWINDOW 'REGION)
                 ;; Position the menu centered directly below the query window, in the space that will later
                 ;; be occupied by the browser
                 (MENU CHOICEMENU (create POSITION
                                       XCOORD _
                                       (+ (fetch (REGION LEFT) of REG)
                                          (IQUOTIENT (- (fetch (REGION WIDTH)
                                                            of REG)
                                                            (fetch IMAGEWIDTH
                                                            of CHOICEMENU))
                                          2))
                                       YCOORD _ (- (fetch (REGION BOTTOM)
                                                         of REG)
                                                  (fetch IMAGEHEIGHT of CHOICEMENU
                                                         )))
                 T]
                (:NEW MASTER)
                (:OLD CACHE)
                (:COPY (LET ((OLDTITLE (WINDOWPROP QFORMWINDOW 'TITLE (CONCAT "Fetching "
                                                                              AR.IDENTIFICATION.STRING
                                                                              " Index"))))
                        (OLDICONFN (WINDOWPROP QFORMWINDOW 'ICONFN (FUNCTION TEXTICON)))
                        W) ; So if you want to shrink the window, you see its state
                        (AR.PROMPT.PRINT QFORMWINDOW "Copying " MASTER "...")
                        [AR.PROMPT.PRINT QFORMWINDOW " finished writing "
                        (SETQ CACHE (COPYFILE MASTER (OR CACHE AR.INDEX.CACHE.FILE.NAME
                                                         ]
                        (WINDOWPROP QFORMWINDOW 'TITLE OLDTITLE)
                        (WINDOWPROP QFORMWINDOW 'ICONFN OLDICONFN)
                        (WINDOWPROP QFORMWINDOW 'ICONWINDOW NIL)
                        CACHE))
                        (SHOULDNT))
                        (CLEARW QFORMWINDOW]
                (WINDOWDELPROP QFORMWINDOW 'CLOSEFN (FUNCTION AR.QFORM.CREATE.ABORT))
                (WINDOWPROP QFORMWINDOW 'PROCESS NIL)]))

```

(AR.QFORM.CREATE.ABORT

```

[LAMBDA (WINDOW)
  (LET [(P (WINDOWPROP WINDOW 'PROCESS))
        (AND P (PROCESSP P)
              (PROCESS.EVAL P ' (ERROR!)))]

```

; Edited 29-Feb-88 15:19 by bvm

(AR.QFORM.GDATE

```

[LAMBDA (DT)
  (if DT

```

; Edited 29-Feb-88 15:21 by bvm

```
then (GDATE DT (DATEFORMAT DAY.OF.WEEK DAY.SHORT NO.SECONDS))
else "Date unknown?"]]
```

(AR.QUERY.WHENSELECTEDFN

```
[LAMBDA (ITEM) ; Edited 1-Mar-88 11:27 by bvm
;; WHENSELECTEDFN for Query and Sort spec button menus. Similar to default, but don't evaluate the cadr.
(if (NLISTP ITEM)
then ITEM
else (CADR ITEM])
```

(AR.QUERY.CLOSEFN

```
[LAMBDA (WINDOW) ; Edited 8-Aug-88 11:10 by bvm
(AR.QUERY.CLOSE/SHRINK WINDOW :CLOSE)]
```

(AR.QUERY.SHRINKFN

```
[LAMBDA (WINDOW) ; Edited 8-Aug-88 11:10 by bvm
(AR.QUERY.CLOSE/SHRINK WINDOW :SHRINK)]
```

(AR.QUERY.CLOSE/SHRINK

```
[LAMBDA (WINDOW HOW) ; Edited 8-Aug-88 11:10 by bvm
;; CLOSEFN or SHRINKFN on Query window: check that we're not busy, then kill the tedit proc, close the index, etc.
(PROG [(BUSYPROC (WINDOWPROP WINDOW 'BROWSER.BUSY)
(if (AND BUSYPROC (PROCESSP BUSYPROC))
then (if (NOT (MOUSECONFIRM (CL:FORMAT NIL "Browser is busy with ~A;
Click LEFT to confirm aborting it.")
T))
then (RETURN 'DON'T)
else (DEL.PROCESS BUSYPROC)))
[if (EQ HOW :SHRINK)
then ; save the textstream
(WINDOWPROP WINDOW 'SAVED-TEXTSTREAM (WINDOWPROP WINDOW 'TEXTSTREAM)
(AR.KILL.ATTACHED.TEDIT.CLOSEFN WINDOW)
(AR.INDEX.FILE.CLOSE WINDOW)
[if (EQ HOW :CLOSE)
then ; Snap link to AR display window
(LET ((W (WINDOWPROP WINDOW 'AR.DISPLAY.WINDOW NIL))
(AND (WINDOWP W)
(WINDOWPROP W 'AR.QUERY.WINDOW NIL))
(RETURN NIL])
```

(AR.QUERY.EXPANDFN

```
[LAMBDA (WINDOW) ; Edited 29-Feb-88 16:29 by bvm
;; On expanding the query window, rebuild the Tedit process behind the query buttons.
(LET ((TS (WINDOWPROP WINDOW 'SAVED-TEXTSTREAM NIL))
(AND TS (AR.INSTALL.TEDITSTREAM WINDOW TS (LIST 'TITLEMENUFN (FUNCTION NIL]))
```

(AR.QFORM.ICONFN

```
[LAMBDA (WINDOW OLDICON) ; Edited 29-Feb-88 16:10 by bvm
(OR OLDICON (ICONW AR.QFORM.ICON NIL (WINDOWPROP WINDOW 'ICONPOSITION))
```

(AR.INDEX.OPEN

```
[LAMBDA (QFORMWINDOW FILENAME) ; Edited 25-Jul-88 15:15 by bvm
;; Open the ar index, setting the variable AR.INDEX.FILE to the stream, and returning the index data
;; The last 32 bits of the file point at the start of the index data.
(PROG (*UPPER-CASE-FILE-NAMES* INDEX.STREAM CONDITION INDEX.DATA)
(AR.PROMPT.PRINT QFORMWINDOW T "Opening index file... ")
[CL:MULTIPLE-VALUE-SETQ (INDEX.STREAM CONDITION)
(IGNORE-ERRORS (OPENSTREAM FILENAME 'INPUT 'OLD)
(if CONDITION
then (AR.PROMPT.PRINT QFORMWINDOW (CL:FORMAT NIL "failed: ~A" CONDITION))
(RETURN NIL))
(replace (STREAM MAXBUFFERS) of INDEX.STREAM with 40)
(SETFILEPTR INDEX.STREAM (- (GETEOFPTR INDEX.STREAM)
BYTESPERCELL))
(SETFILEPTR INDEX.STREAM (\DWIN INDEX.STREAM))
(SETQ INDEX.DATA (READ INDEX.STREAM FILERDTBL))
(COND
((NOT (type? AR.INDEX.DATA INDEX.DATA))
(CLOSEF INDEX.STREAM)
(AR.PROMPT.PRINT QFORMWINDOW "failed: Bad index format")
(RETURN NIL))
(replace (AR.INDEX.DATA AR.INDEX.FILE) of INDEX.DATA with INDEX.STREAM)
[if (NOT (fetch (AR.INDEX.DATA AR.MAX.INDEX) of INDEX.DATA))
then ; Max.index not normally stored in file, we derive it
(SETQ INDEX.DATA (create AR.INDEX.DATA using INDEX.DATA AR.MAX.INDEX
(SUB1 (IQUOTIENT (- (fetch (AR.INDEX.DATA
```



```

                                substring")
                                ,@(AR.QLIST.MENU.COMPARISONS FIELD.NAME]
T)
[AND AR.QUERY.SPEC.ITEMS (LIST `("Special" NIL "Select custom queries from submenu"
                                (SUBITEMS ,@AR.QUERY.SPEC.ITEMS)
'(("--Clear--" :CLEAR "Clear the Query spec and start over" ]
WHENSELECTEDFN _ (FUNCTION AR.QUERY.WHENSELECTEDFN)
CENTERFLG _ T))
(WINDOWPROP QFORMWINDOW 'AR.QLIST.PROMPT.MENU VAL)
VAL]]

```

(AR.QLIST.MENU.COMPARISONS

```

[LAMBDA (FIELD.NAME) ; Edited 16-Mar-88 17:16 by bvm
;; Return a set of menu items for arithmetic comparisons on FIELD.NAME
(for OP in AR.COMPARISON.OPERATORS collect (LIST OP (LIST FIELD.NAME OP]))

```

(AR.QFORM.PROMPT.LIST.FN

```

[LAMBDA (OBJ SEL WINDOW) ; Edited 20-Jul-88 15:56 by bvm
(LET*

```

```

(*PACKAGE* *INTERLISP-PACKAGE*)
(TOBJ (fetch (SELECTION \TEXTOBJ) of SEL))
(WINDOW (CAR (fetch (TEXTOBJ \WINDOW) of TOBJ)))
(OPERATION (IMAGEOBJPROP OBJ 'MBTEXT))
[ITEM (MENU (SELECTQ OPERATION
(|Query List:|
(AR.GET.QLIST.PROMPT.MENU WINDOW))
(|Sort List:| (AR.GET.SLIST.PROMPT.MENU WINDOW))
(SHOULDNT)
ISP) ; Set cursor back to point at button
[if ITEM
then
(PROG ((FIELD.SEL (MBUTTON.FIND.NEXT.FIELD TOBJ (fetch (SELECTION CH#) of SEL)))
FIELD.END.CH#)
(if (EQ ITEM :CLEAR)
then (EDIT.DELETE TOBJ FIELD.SEL)
else [if (NOT (STRINGP ITEM))
then ; Turn the item into something that can be read back by query reader, which uses FILERDTBL.
(SETQ ITEM
(if (OR (NLISTP ITEM)
(LISTP (CDDR ITEM)))
then (MKSTRING (OR ITEM (RETURN))
T FILERDTBL)
elseif (EQ (CAR ITEM)
'FUNCTION)
then ; Computed item
(SETQ ITEM (CL:FUNCALL (CADR ITEM)
WINDOW))
(if (NULL ITEM)
then (RETURN)
elseif (STRINGP ITEM)
else (if (AND (LISTP ITEM)
(EQ OPERATION ' |Sort List:|))
then ; Strip off parens for sort spec
(SUBSTRING (MKSTRING ITEM T FILERDTBL)
2 -2)
else (MKSTRING ITEM T FILERDTBL)))
elseif (EQ (CADR ITEM)
'HAS)
then ; Substring search
(CONCAT "(" (MKSTRING (CAR ITEM)
T FILERDTBL)
" HAS >>string<<")
elseif (MEMB (CADR ITEM)
AR.COMPARISON.OPERATORS)
then (DESTRUCTURING-BIND (FIELD.NAME OP)
ITEM
(LET [(TEMPLATE (if (EQ FIELD.NAME 'Number%)
then
; no quotes
(LIST ">>" "num<<")
else
; Have to quote dates
(LIST "%>>" "date<<%")
; Comparison (fieldname op valuetype)
(CONCAT "(" (MKSTRING FIELD.NAME T FILERDTBL)
(if (EQ OP 'btwn)
then
; E.g., (fieldname >= >>lonum<< <= >>hinum<<)
(CONCAT (AR.MAKE.COMPARISON.STRING
'>= TEMPLATE)
(AR.MAKE.COMPARISON.STRING
'<= TEMPLATE))
else (AR.MAKE.COMPARISON.STRING OP TEMPLATE))

```

```

                    " ) " ) ) )
else ; Specific value search
  (SETQ ISP (MKSTRING (LIST (CAR ITEM)
                            ' IS
                            (CADR ITEM)
                            T FILERDTBL]
  (TEDIT.INSERT TOBJ ITEM (SETQ FIELD.END.CH# (+ (fetch (SELECTION CH#) of FIELD.SEL)
                                                (fetch (SELECTION DCH) of FIELD.SEL]
  (TEDIT.INSERT TOBJ " " (+ FIELD.END.CH# (NCHARS ITEM)))
  (COND
    ((STRPOS ">>" ITEM)
     (TEDIT.SETSEL TOBJ FIELD.END.CH# 0)
     (TEDIT.NEXT TOBJ))
    (ISP ; Delete-select the IS so you can change it to , say, >
     (TEDIT.SETSEL TOBJ (+ FIELD.END.CH# (STRPOS " IS " ITEM))
                        2 NIL T]
  (CURSORPOSITION (create POSITION
                          XCOORD _ 20
                          YCOORD _ (DSPYPOSITION NIL WINDOW))
  WINDOW])

```

(AR.QFORM.TITLEMENU

```

[LAMBDA (WINDOW) ; Edited 20-Jul-88 16:12 by bvm
  (LET [(OP (MENU (OR AR.QFORM.TITLEMENU (SETQ AR.QFORM.TITLEMENU (create MENU
                                                                    ITEMS _ (REMOVE 'btwn
                                                                    AR.COMPARISON.OPERATORS
                                                                    )
                                                                    CENTERFLG _ T]
  (if OP ; Type this into the window
    then (TEDIT.INSERT (TEXTSTREAM WINDOW)
                       OP])

```

(AR.MAKE.COMPARISON.STRING

```

[LAMBDA (OP TEMPLATE) ; Edited 16-Mar-88 17:11 by bvm
  (CONCAT " " OP " " (CAR TEMPLATE)
  (SELECTQ OP
    ((> >=)
     "lo")
    ((< <=)
     "hi")
    ("")
  (CADR TEMPLATE])

```

(AR.GET.BUTTON.FIELD.AS.LIST

```

[LAMBDA (FORMWINDOW FIELD.NAME) ; Edited 24-Feb-88 21:10 by bvm
;; READ, using FILERDTBL, the value of FIELD.NAME of FORMWINDOW, returning a list.
(LET* ((TOBJ (TEXTOBJ FORMWINDOW))
       (BUTTON (AR.FIND.BUTTON TOBJ FIELD.NAME))
       (FIELD.VAL (MBUTTON.NEXT.FIELD.AS.TEXT TOBJ (CDR BUTTON)))
       (STREAM (OPENSTRINGSTREAM FIELD.VAL))
       (*PARENS* 0)
       *MAX-PARENS*)
  [SETFILEINFO STREAM 'ENDOFSTREAMOP (FUNCTION (LAMBDA (STREAM)
                                                ; Handler for eof error. We try adding some closing parens
                                                (if (NULL *MAX-PARENS*)
                                                    then (SETQ *MAX-PARENS* (CL:COUNT #\ ( FIELD.VAL)))
                                                    (if (> (add *PARENS* 1)
                                                        *MAX-PARENS*)
                                                        then
                                                        ; Let's not try to add more close parens than open
                                                        (AR.PROMPT.PRINT FORMWINDOW T "Malformed "
                                                            FIELD.NAME ", command aborted")
                                                        (ERROR!))
                                                        ; return a closing paren
                                                        (CHARCODE " ) ")]
  (bind X (*READTABLE* _ FILERDTBL) until (EQ (SETQ X (CL:READ STREAM NIL STREAM))
                                               STREAM)
  collect X finally (if (> *PARENS* 0)
                       then ; We had to add some right parens to make it balance, so fix the
                               ; button field.
                               (TEDIT.INSERT TOBJ (ALLOCSTRING *PARENS* (CHARCODE " ) ")))
                       (+ (CDR BUTTON)
                          (NCHARS FIELD.VAL)
                          1])
)

```

;; AR Browser window stuff

(DEFINEQ

(AR.BROWSER.PRINTFN

```
[LAMBDA (BROWSER ITEM WINDOW) ; Edited 15-Jun-90 11:06 by jds
  ;; Repaint the line in the Query browser window corresponding to this AR.
  (LET ((ENTRY.DATA (fetch TIDATA of ITEM))
        (STREAM (GETSTREAM WINDOW))
        (MAINW (MAINWINDOW WINDOW)))
    (if (NOT (fetch (ARQUERYDATA ARQCOMPLETE) of ENTRY.DATA))
        then (AR.ENSURE.QUERY.DATA.ITEM MAINW ENTRY.DATA))
    ;; The fields in ENTRY.DATA contain either a value or a (ptr length) pair for string fields. The specs in AR.DISPLAY.FIELDS give (fieldname
    ;; desiredwidth). We place 2 spaces between fields.
    (PRINTOUT WINDOW .I5 (fetch (ARQUERYDATA ARQ#) of ENTRY.DATA)
      " ")
    (for SPEC in (WINDOWPROP MAINW 'AR.DISPLAY.FIELDS) as VALUE in (fetch (ARQUERYDATA ARQFIELDS) of
      ENTRY.DATA
    )
      bind WIDTH LEN (SCRATCH _ (WINDOWPROP MAINW 'AR.FORM.SCRATCH.STREAM))
      do (SETQ WIDTH (CADR SPEC))
        (if (NOT VALUE)
            then (SPACES (+ WIDTH 2)
              STREAM)
            elseif (LISTP VALUE)
              then ; bits are on scratch file
                (SETFILEPTR SCRATCH (CAR VALUE))
                (COPYBYTES SCRATCH STREAM (SETQ LEN (MIN (CADR VALUE)
                  WIDTH)))
                (SPACES (- (+ WIDTH 2)
                  LEN)
                  STREAM)
            else ; VALUE is it
              (AR.PRINT.PADED VALUE STREAM 1 WIDTH (+ WIDTH 2]))
```

(AR#.FROM.ITEM

```
[LAMBDA (ITEM QFORMWINDOW) ; Edited 15-Jun-90 11:06 by jds
  (LET ((DATA (fetch TIDATA of ITEM)))
    (if (NOT (fetch (ARQUERYDATA ARQCOMPLETE) of DATA))
        then (AR.ENSURE.QUERY.DATA.ITEM QFORMWINDOW DATA))
    (fetch (ARQUERYDATA ARQ#) of DATA))
```

(AR.BROWSER.COMMANDFN

```
[LAMBDA (ITEM MENU BUTTON) ; Edited 20-Jul-88 18:42 by bvm
  ;; WHENSELECTEDFN for the AR query browser menu. We spawn a process to do the work
  (LET* [(MAINW (MAINWINDOW (WFORMMENU MENU)))
        (BROWSERW (WINDOWPROP MAINW 'QFORM.ENTRY.WINDOW))
        (if BROWSERW
            then (LET ((BROWSER (WINDOWPROP BROWSERW 'TABLEBROWSER))
                      (FN (CADR ITEM)))
              (if (if (NLISTP FN)
                  then ; Normal case, require that there be something in the
                  ; tablebrowser
                    BROWSER
                  else ; Do it anyway, e.g., Query
                    (SETQ FN (CAR FN)))
                then (ADD.PROCESS `((FUNCTION AR.BROWSER.DO.COMMAND)
                  ',MAINW
                  ',BROWSER
                  ',FN
                  ',ITEM
                  ',MENU
                  ',BUTTON)
                  'NAME
                  (CONCAT "AR-" (CAR ITEM))
                  'BEFOREEXIT
                  'DON'T)
                else (AR.PROMPT.PRINT MAINW :CLEAR "There are no ARs in the browser."))
```

(AR.BROWSER.DO.COMMAND

```
[LAMBDA (WINDOW BROWSER FN ITEM MENU BUTTON) ; Edited 4-Aug-88 11:14 by bvm
  ;; Started up in its own process to perform the action specified by the menu item. Menu functions get called with arglist (window browser button).
  [if (NOT (MEMBER ITEM (fetch (MENU ITEMS) of MENU)))
      then ; Subitem--shade the main item
        (SETQ ITEM (find I in (fetch (MENU ITEMS) of MENU) suchthat (MEMBER ITEM (CDR (CADDR I))
        (if (WINDOWPROP WINDOW 'BROWSER.BUSY (THIS.PROCESS))
            then (TB.BROWSER.BUSY BROWSER)
            else [RESETSAVE NIL (LIST [FUNCTION (LAMBDA (WINDOW ITEM MENU PROCNAME)
              (WINDOWPROP WINDOW 'BROWSER.BUSY NIL)
              (SHADEITEM ITEM MENU WHITESHAE)
              (if [AND PROCNAME (EQ PROCNAME (PROCESSPROP (TTY.PROCESS)
                'NAME)
                then ; Give the tty back, unless someone has already taken it
                  (TTY.PROCESS (WINDOWPROP WINDOW 'PROCESS])
```

```

WINDOW ITEM MENU (LET [(PROC (WINDOWPROP WINDOW 'PROCESS)]
  (if (AND PROC (TTY.PROCESSP PROC))
    then ; Take TTY away from query window so that cursor isn't flashing
        ;there
        (TTY.PROCESS 'BACKGROUND)
        (PROCESSPROP (TTY.PROCESS)
          'NAME]
(SHADEITEM ITEM MENU AR.WHENSELECTEDSHADE)
(AR.PROMPT.CLEAR WINDOW)
(CL:FUNCALL FN WINDOW BROWSER BUTTON])

```

(AR.BROWSER.SELECTED.ARS

[LAMBDA (WINDOW BROWSER LASTPROP) ; Edited 17-Feb-88 14:04 by bvm

:: Return list of items selected in BROWSER. If LASTPROP is specified, choose exactly one of the items, based on the idea that the item in window's LASTPROP property was most recently accessed, so if that one is selected, go on to the next. Returns NIL if no selected item, or selection has run out.

```

(LET [(SELECTEDARS (TB.COLLECT.ITEMS BROWSER 'SELECTED)]
  (if (NULL SELECTEDARS)
    then (AR.PROMPT.PRINT WINDOW "No AR is selected")
        NIL
    elseif (NULL LASTPROP)
    then ; return them all
        SELECTEDARS
    else (PROG ((LASTITEM (WINDOWPROP WINDOW LASTPROP))
              (ITEM NEXTITEM TAIL))
      (if (NULL (CDR SELECTEDARS))
        then ; Only one selected, so choose that one item, or go on to the next if that one was most recently
              ; displayed/edited.
              (if (EQ (SETQ ITEM (CAR SELECTEDARS))
                LASTITEM)
                then ; Advance selection to next item
                    (if [SETQ NEXTITEM (TB.NTH.ITEM BROWSER (ADD1 (fetch TI# of ITEM)]
                  then (TB.UNSELECT.ITEM BROWSER ITEM)
                      (TB.SELECT.ITEM BROWSER (SETQ ITEM NEXTITEM))
                  else (RETURN (AR.PROMPT.PRINT WINDOW "That was the last AR")
                          ; Cycle thru a group of selected ars.
                          (if (NULL (SETQ TAIL (MEMB LASTITEM SELECTEDARS)))
                            then ; None recently displayed, so show the first
                                (SETQ ITEM (CAR SELECTEDARS))
                            elseif (CDR TAIL)
                            then ; Choose the next
                                (SETQ ITEM (CADR TAIL))
                            else (WINDOWPROP WINDOW LASTPROP NIL)
                                (RETURN (AR.PROMPT.PRINT WINDOW "That was the last selected AR" T "Click again
                                  to cycle back to the first one."])
                                (RETURN ITEM]))

```

(AR.BROWSER.DISPLAY

[LAMBDA (WINDOW BROWSER BUTTON) ; Edited 4-Aug-88 14:42 by bvm

:: Displays the selected AR in a readonly display window

```

(LET ((ITEM (AR.BROWSER.SELECTED.ARS WINDOW BROWSER 'LAST.DISPLAYED.AR))
  DISPLAYW)
  (if ITEM
    then (TB.NORMALIZE.ITEM BROWSER ITEM) ; Scroll so visible, if necessary
        (if [OR (EQ BUTTON 'MIDDLE)
          (NOT (WINDOWP (SETQ DISPLAYW (WINDOWPROP WINDOW 'AR.DISPLAY.WINDOW))
            ; Make a display window
            (SETQ DISPLAYW (CREATEW (REGIONP DISPLAYW)
              (CONCAT AR.IDENTIFICATION.STRING " display window"))))
          (WINDOWPROP DISPLAYW 'ICONFN (FUNCTION TEXTICON))
          (if (NEQ BUTTON 'MIDDLE)
            then ; Remember it for next time
                (WINDOWPROP WINDOW 'AR.DISPLAY.WINDOW DISPLAYW)
                (WINDOWPROP DISPLAYW 'AR.QUERY.WINDOW WINDOW)
                (WINDOWADDDPROP DISPLAYW 'CLOSEFN (FUNCTION (LAMBDA (WINDOW)
                  ; When display window is closed, make the query window
                  ; remember only the region
                  (LET
                    ((Q (WINDOWPROP WINDOW
                      'AR.QUERY.WINDOW NIL)
                    )))
                    (AND Q (WINDOWPROP Q
                      'AR.DISPLAY.WINDOW
                      (WINDOWREGION
                        WINDOW])
                (AR.DISPLAY (AR#.FROM.ITEM ITEM WINDOW)
                  DISPLAYW)
                (WINDOWPROP WINDOW 'LAST.DISPLAYED.AR ITEM]))

```

(AR.BROWSER.EDIT

```

[LAMBDA (WINDOW BROWSER BUTTON) ; Edited 5-Aug-88 11:07 by bvm
  ;; Edits the selected AR in an AReedit window
  (PROG ((ITEM (AR.BROWSER.SELECTED.ARS WINDOW BROWSER 'LAST.EDITED.AR))
    EDITW TOBJ MENUW)
    (if (NULL ITEM)
      then (RETURN))
    (TB.NORMALIZE.ITEM BROWSER ITEM) ; Scroll so visible, if necessary
    (if (EQ BUTTON 'MIDDLE)
      then ; Always get a new window and don't hang onto it
        elseif [AND [WINDOWP (SETQ EDITW (WINDOWPROP WINDOW 'AR.EDIT.WINDOW)
          (OR (OPENWP EDITW)
            (OPENWP (WINDOWPROP EDITW 'ICONWINDOW)
              ; Use this window--it's still open, or is shrunk
            else (AR.PROMPT.PRINT WINDOW "Select AR Edit window to use" T "or click >>here<< to make new
              window.")
              (SETQ EDITW (WHICHW (GETPOSITION)))
              (AR.PROMPT.CLEAR WINDOW)
              (if (NULL EDITW)
                then (GO ABORT)
                elseif (EQ EDITW (GETPROMPTWINDOW WINDOW)) ; Want to make a new one
                  then (SETQ EDITW NIL)
                  elseif (EQ (WINDOWPROP (SETQ EDITW (OR (WINDOWPROP EDITW 'MAINWINDOW)
                    (WINDOWPROP EDITW 'ICONFOR)
                      EDITW))
                      'AR.WINDOW.PROC.NAME)
                    'AR.FORM)
                    then (WINDOWPROP WINDOW 'AR.EDIT.WINDOW EDITW)
                    else (GO ABORT)))
    (if EDITW
      then ; Check to see that the specified window is useable
        (SETQ MENUW (AR.GET.MENU.FROM.MAIN.WINDOW EDITW))
        (if (OR [NULL (SETQ TOBJ (WINDOWPROP MENUW 'TEXTOBJ])
          (AR.TOBJ.ACTIVEP TOBJ)
          [NULL (SETQ TOBJ (WINDOWPROP EDITW 'TEXTOBJ])
            (AR.TOBJ.ACTIVEP TOBJ))
          then (if (NOT (MOUSECONFIRM "That window is busy, click LEFT to get new window" T
            (GETPROMPTWINDOW WINDOW)))
            then (GO ABORT))
            (SETQ EDITW NIL)
          elseif (TEDIT.STREAMCHANGEDP (TEXTSTREAM EDITW))
            then (AR.PROMPT.PRINT WINDOW T "The form in that window has not been saved." T "Click LEFT
              to confirm smashing it anyway.")
              (if (MOUSECONFIRM T T (GETPROMPTWINDOW WINDOW)
                T)
                then ; Mark stream unchanged so the Get will proceed
                  (AR.PROMPT.CLEAR WINDOW)
                  (TEDIT.STREAMCHANGEDP (TEXTSTREAM EDITW)
                    T)
                  elseif (MOUSECONFIRM "Do you want to use a new window?" NIL (GETPROMPTWINDOW WINDOW))
                    then (SETQ EDITW NIL)
                    else (GO ABORT])
    (if EDITW
      then ; Still have a window to play with
        (if (NOT (OPENWP EDITW))
          then ; Explicitly open it before Get to avoid some attached window
            ; glitches. TEdit expandfn grabs tty, so give it back
            (EXPANDW (WINDOWPROP EDITW 'ICONWINDOW)
              (TTY.PROCESS T))
            (AR.FORM.PROGRAMMATIC.GET MENUW (AR#.FROM.ITEM ITEM WINDOW))
          else (SETQ EDITW (AR.FORM.GROUP.CREATE (AR#.FROM.ITEM ITEM WINDOW))
            (if (NEQ BUTTON 'MIDDLE)
              then ; Remember it for next time
                (WINDOWPROP WINDOW 'AR.EDIT.WINDOW EDITW))
            (WINDOWPROP WINDOW 'LAST.EDITED.AR ITEM) ; Mark this item as both edited and displayed
            (RETURN (WINDOWPROP WINDOW 'LAST.DISPLAYED.AR ITEM))
    ABORT
    (AR.PROMPT.PRINT WINDOW T "Command aborted"]])

```

(AR.BROWSER.HARDCOPY

```

[LAMBDA (WINDOW BROWSER BUTTON) ; Edited 4-Aug-88 14:42 by bvm
  (LET ((ARS (AR.BROWSER.SELECTED.ARS WINDOW BROWSER)))
    (if [AND ARS (SETQ ARS (AR.HARDCOPY (for X in ARS collect (AR#.FROM.ITEM X WINDOW))
      (GETPROMPTWINDOW WINDOW)
      then (AR.PROMPT.PRINT WINDOW T "Done, " ARS])

```

)

;; Sorting

(DEFINEQ

(AR.QFORM.SORT

```

[LAMBDA (QFORMWINDOW TBROWSER) ; Edited 22-Jul-88 16:48 by bvm

```



```

[AND AR.SORT.SPEC.ITEMS (LIST `("Special" NIL "Select custom sort
                                orders from submenu"
                                (SUBITEMS ,@AR.SORT.SPEC.ITEMS]
                                `(("--Clear--" :CLEAR "Clear the Sort spec and start over")
                                CENTERFLG _ T
                                WHENSELECTEDFN _ (FUNCTION AR.QUERY.WHENSELECTEDFN]
(WINDOWPROP QFORMWINDOW 'AR.SLIST.PROMPT.MENU VAL)
VAL])

```

(AR.ENSURE.QUERY.FIELDS

```

[LAMBDA (QFORMWINDOW FIELDS) ; Edited 29-Feb-88 11:50 by bvm
;; Ensures that all the entries in the query window have the specified fields. If not, we fetch them. Returns the complete list of fields stored in the
;; entries.
(AR.AUGMENT.QUERY.FIELDS QFORMWINDOW (for F in FIELDS bind (KNOWN _ (WINDOWPROP QFORMWINDOW
                                'AR.FIELD.DESCRPTIONS))
                                collect F unless (ASSOC F KNOWN)))
(AR.ENSURE.QUERY.DATA QFORMWINDOW FIELDS])

```

(AR.ENSURE.QUERY.DATA

```

[LAMBDA (QFORMWINDOW FIELDS ENTRIES) ; Edited 15-Jun-90 11:09 by jds
;; Makes sure that all of FIELDS are filled in for ENTRIES, or all entries in window if NIL.
(PROG* ((INDEX.STREAM AR.INDEX.FILE)
        (DESCRS (WINDOWPROP QFORMWINDOW 'AR.FIELD.DESCRPTIONS))
        (NFIELDS (LENGTH DESCRS))
        SCRATCH)
[OR ENTRIES (SETQ ENTRIES (WINDOWPROP QFORMWINDOW 'AR.ENTRIES]
(if (for E in ENTRIES always (fetch (ARQUERYDATA ARQCOMPLETE) of E))
    then ; Nothing to do
        (RETURN))
(if [for ENTRY in (CDR ENTRIES) bind (LASTINDEX _ (CAAR ENTRIES)) thereis (> LASTINDEX (SETQ LASTINDEX
                                (CAR ENTRY))
                                then
    ;; Entries are out of order (e.g., we have been called to fetch fields to print or to resort). More efficient for us to work in sorted
    ;; order, so make ourselves a sorted copy. Only copy the first cons, so that our destructive changes to the tail will affect
    ;; original ENTRIES, too.
    (SETQ ENTRIES (SORT (for ENTRY in ENTRIES collect (CONS (CAR ENTRY)
                                (CDR ENTRY)))
                        T)))

```

;; Gather the data in two passes: first, scan the table of fixed-size entries, gathering AR numbers plus the address info of string fields. In the
;; second pass, scan the region associated with each field, collecting values. This results in better locality of reference in the index file.

```

(if [OR (EQ FIELDS T)
        (for FIELD in FIELDS thereis (FIXP (fetch (ARINDEXDESCR ARIOFFKEYS) of (CL:ASSOC FIELD DESCRS]
    then
    ;; There are some string fields to fill in, might as well get them all
    ;; (since they all live in the same place)
    ;; Need scratch stream to store the values (could store as strings,
    ;; but that's more expensive)
    (SETFILEPTR (SETQ SCRATCH (AR.GET.SCRATCH.STREAM QFORMWINDOW))
                -1)
    (SETFILEPTR INDEX.STREAM AR.INDEX.ENTRY.BEGIN.PTR)
    ;; Start of fixed-size entries
    (for ENTRY in ENTRIES bind (LASTOFFSET _ 0)
                                (LASTINDEX _ 0)
                                INDEX
    unless (fetch (ARQUERYDATA ARQCOMPLETE) of ENTRY)
    do
    ;; This loop goes entry by entry. Loop invariant is that the file is positioned at LASTOFFSET of LASTINDEX. Collect
    ;; pointers for string fields, ignore enumerated fields.
    [SETQ LASTOFFSET (AR.COLLECT.ENTRY.FIELDS
                                ENTRY DESCRS (PROGN ; Relative to this new INDEX, the LASTOFFSET that we read at
                                ; is this much farther back.
                                (- LASTOFFSET
                                (TIMES (- (SETQ INDEX (fetch (ARQUERYDATA
                                ARQINDEX)
                                of ENTRY))
                                LASTINDEX)
                                AR.INDEX.ENTRY.SIZE]
    (SETQ LASTINDEX INDEX)))

```

;; At this point, each of ENTRIES is filled with values of either NIL for enumerated fields or (offset length) for variable fields. Now take a field at a
;; time, and fill in the real values.

```

(for D in DESCRS as I from 0 bind BEGIN VALUES LASTINDEX [FINALI _ (AND (EQ FIELDS T)
                                (SUB1 (LENGTH DESCRS]
when (AND (SETQ BEGIN (fetch (ARINDEXDESCR ARIBEGIN) of D))
          (OR (EQ FIELDS T)
              (CL:MEMBER (fetch (ARINDEXDESCR ARINAME) of D)
                          FIELDS)))
do (if (SETQ VALUES (LISTP (fetch (ARINDEXDESCR ARIOFFKEYS) of D)))
    then
    ;; For enumerated fields, we'll continue to use the incfileptr trick.
    ;; File is always positioned to read the byte for LASTINDEX,
    ;; which is also bytecount relative to BEGIN
    (SETFILEPTR INDEX.STREAM BEGIN)
    (SETQ LASTINDEX 0))

```

```

(for ENTRY in ENTRIES bind PAIR TAIL KEY unless (fetch (ARQUERYDATA ARQCOMPLETE) of ENTRY)
do [if (NULL (CDR ENTRY))
then
; First time for this guy, get some space
(RPLACD ENTRY (CONS NIL (to NFIELDS collect '?])
(SETQ TAIL (CL:NTHCDR I (fetch (ARQUERYDATA ARQALLFIELDS) of ENTRY)))
; (CAR TAIL) is where we want the value
[if VALUES
then
; Get byte for enumerated value
[if (EQ (CAR TAIL)
'?))
then (\INCFILEPTR INDEX.STREAM (- (CAR ENTRY)
LASTINDEX))
(SETQ LASTINDEX (ADD1 (CAR ENTRY)))
(RPLACA TAIL (COND
((NEQ (SETQ KEY (BIN INDEX.STREAM)
0) ; Zero denotes the null value
(CL:NTH (SUB1 KEY)
VALUES]
elseif (AND (SETQ PAIR (CAR TAIL))
(< (CAR PAIR)
0))
then
; String field--PAIR is (-offset-1 length). Copy its contents to the
; scratch stream, and replace the offset pointer with the scratch
; file ptr
(SETFILEPTR INDEX.STREAM (- BEGIN (CAR PAIR)
1) ; i.e. (+ begin (- -1 (car pair)))
(RPLACA PAIR (PROG1 (GETFILEPTR SCRATCH)
(COPYBYTES INDEX.STREAM SCRATCH (CADR PAIR))))]
(if (EQ I FINALI)
then
; All done now
(replace (ARQUERYDATA ARQCOMPLETE) of ENTRY with T])

```

(AR.COLLECT.ENTRY.FIELDS

```

[LAMBDA (ENTRY DESCRS LASTOFFSET) ; Edited 15-Jun-90 11:09 by jds
;; Fill in the "Fixed-size" entry fields in ENTRY, an item from an AR query browser. DESCRS is the description list, paralleling the ALLFIELDS tail
;; of ENTRY. LASTOFFSET is the offset past the last entry read from the file, relative to this entry. We smash ENTRY and return a new
;; LASTOFFSET.
[if (NULL (CDR ENTRY))
then
; First time for this guy, get some space
(RPLACD ENTRY (CONS NIL (for D in DESCRS collect '?])
[for D in DESCRS as TAIL on (fetch (ARQUERYDATA ARQALLFIELDS) of ENTRY)
bind (INCREMENT _ AR.INDEX.ENTRY.SIZE)
(STREAM _ AR.INDEX.FILE)
(MAX.INDEXP _ (EQ (fetch (ARQUERYDATA ARQINDEX) of ENTRY)
AR.MAX.INDEX))
OFFSET LEN VALUE
when (AND (FIXP (SETQ OFFSET (fetch (ARINDEXDESCR ARIOFFKEYS) of D)))
(EQ (CAR TAIL)
'?))
do
; This is a field stored in the fixed-size entry table. Bump the
; fileptr to the next spot. We use \incfileptr to avoid creating
; number boxes.
; Equivalent to (setfileptr index.stream (ar.entry.ptr.from.index
; index offset))
(INCFILEPTR STREAM (- OFFSET LASTOFFSET))
(SETQ LASTOFFSET (+ OFFSET AR.BYTES.PER.PTR))
(SETQ VALUE (\DWIN STREAM))
(RPLACA TAIL (if (EQ OFFSET 0)
then
; We just read the AR number, that's all we need to do.
VALUE
else
; We just read the offset of the field data. Need to get the offset
; of the next AR's field in order to compute the length
(SETQ LEN (- (if MAX.INDEXP
then
; There is no next one, so all we know is this field goes to the
; end. Sure would have been nice to have a dummy n+1 entry.
(- (fetch (ARINDEXDESCR ARIEND) of D)
(fetch (ARINDEXDESCR ARIBEGIN) of D))
else (\INCFILEPTR STREAM (- INCREMENT AR.BYTES.PER.PTR))
(add LASTOFFSET INCREMENT)
; We have bumped file pointer exactly one index forward.
(\DWIN STREAM))
VALUE))
; For now, we return (-offset-1 length), unless length is 0, in
; which case the field is empty. The extra -1 is because offset
; can be zero.
(AND (NEQ LEN 0)
(LIST (- -1 VALUE)
LEN]
LASTOFFSET])

```

(AR.ENSURE.QUERY.DATA.ITEM

```

[LAMBDA (QFORMWINDOW ENTRY) ; Edited 15-Jun-90 11:09 by jds
;; Fill in all the fields of one particular ENTRY. This is a relatively inefficient operation, since we duplicate effort used by reading data for the other
;; entries, and we read the file in a suboptimal order. It exists solely for the PRINTFN, and is hacked specially. This code is coordinated with
;; AR.ENSURE.QUERY.DATA.

```

```

(RESETLST ; Code begins with a manual WITH.AR.QUERY here...
  (LET [(LOCK (WINDOWPROP QFORMWINDOW 'AR.INDEX.MONITORLOCK))
        (INDEX.DATA (WINDOWPROP QFORMWINDOW 'AR.INDEX.DATA))
        (if (NOT (OBTAIN.MONITORLOCK LOCK T T))
            then ; Lock is in use. Don't steal mouse
              (AR.PROMPT.PRINT QFORMWINDOW " [Browser busy; please wait] ")
              (ALLOW.BUTTON.EVENTS)
              (OBTAIN.MONITORLOCK LOCK NIL T))
        (LET ((AR.INDEX.FILE (fetch (AR.INDEX.DATA AR.INDEX.FILE) of INDEX.DATA))
              (AR.INDEX.ENTRY.SIZE (fetch (AR.INDEX.DATA AR.INDEX.ENTRY.SIZE) of INDEX.DATA))
              (AR.INDEX.ENTRY.BEGIN.PTR (fetch (AR.INDEX.DATA AR.INDEX.ENTRY.BEGIN.PTR) of INDEX.DATA))
              (AR.MAX.INDEX (fetch (AR.INDEX.DATA AR.MAX.INDEX) of INDEX.DATA))
              (DESCRS (WINDOWPROP QFORMWINDOW 'AR.FIELD.DESCRPTIONS))
              (SCRATCH (WINDOWPROP QFORMWINDOW 'AR.FORM.SCRATCH.STREAM))
              (INDEX (fetch (ARQUERYDATA ARQINDEX) of ENTRY)))
          ;; Note: We Know that only these first 4 variables are needed by the code that follows, not the whole set
          (if (NOT (OPENP AR.INDEX.FILE))
              then (AR.INDEX.FILE.REOPEN QFORMWINDOW))
          (SETFILEPTR SCRATCH -1)
          (SETFILEPTR AR.INDEX.FILE (AR.ENTRY.PTR.FROM.INDEX INDEX))
          (AR.COLLECT.ENTRY.FIELDS ENTRY DESCRS 0) ; Position at start of item
          ; Get fixed table items
          [for D in DESCRS as TAIL on (fetch (ARQUERYDATA ARQALLFIELDS) of ENTRY)
            bind BEGIN VALUES PAIR KEY when (SETQ BEGIN (fetch (ARINDEXDESCR ARIBEGIN) of D))
            do (if (LISTP (SETQ VALUES (fetch (ARINDEXDESCR ARIOFFKEYS) of D)))
                then ; Enumerated field--read a byte and translate it. VALUES is the
                  ; list of keys
                  [if (EQ (CAR TAIL)
                        '?)
                    then (SETFILEPTR AR.INDEX.FILE (+ BEGIN INDEX))
                        (RPLACA TAIL (COND
                            ((NEQ (SETQ KEY (BIN AR.INDEX.FILE))
                                0)
                             ; Zero denotes the null value
                             (CL:NTH (SUB1 KEY)
                                     VALUES)
                            ))
                    elseif (AND (SETQ PAIR (CAR TAIL))
                                (< (CAR PAIR)
                                    0))
                        then ; String field--PAIR is (-offset-1 length). Copy its contents to the
                          ; scratch stream, and replace the offset pointer with the scratch
                          ; file ptr
                          (SETFILEPTR AR.INDEX.FILE (- BEGIN (CAR PAIR)
                                                         1))
                          ; i.e. (+ begin (- -1 (car pair)))
                          (RPLACA PAIR (PROG1 (GETFILEPTR SCRATCH)
                                              (COPYBYTES AR.INDEX.FILE SCRATCH (CADR PAIR))))]
                  (replace (ARQUERYDATA ARQCOMPLETE) of ENTRY with T))))])

```

(AR.AUGMENT.QUERY.FIELDS

[LAMBDA (QFORMWINDOW FIELDS) ; Edited 15-Jun-90 11:07 by jds

;; Add FIELDS to the set of field info stored in the entries in QFORMWINDOW

```

(LET [(NFIELDS (LENGTH FIELDS))
      (DESCRS (for FIELD.NAME in FIELDS collect (LET* ((FIELD.SPEC (CDR (ASSOC FIELD.NAME AR.INDEX.FIELD.SPECS)
                                          ))
                                                       (FIELD.OFFSET (LISTGET FIELD.SPEC 'FIELD.OFFSET))
                                                       KEYS)
          (create ARINDEXDESCR
                 ARINAME _ FIELD.NAME
                 ARIOFFKEYS _ (if FIELD.OFFSET
                                   elseif (SETQ KEYS (LISTGET
                                                         FIELD.SPEC
                                                         '
                                                         ENUMERATED.FIELD.KEYLIST
                                                         ))
                                   then ; Turn this plist into a simple sorted list
                                   (AR.KEYVALS.FROM.KEYLIST
                                    KEYS)
                                   else (HELP "Field for display is
                                             neither string nor
                                             enumerated" FIELD.NAME))
                 ARIBEGIN _ (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR)
                 ARIEND _ (AND FIELD.OFFSET (LISTGET FIELD.SPEC
                                                         'FIELD.END.PTR)
                             ))
          (OLDDDESCRS (WINDOWPROP QFORMWINDOW 'AR.FIELD.DESCRPTIONS))
          (ENTRIES (WINDOWPROP QFORMWINDOW 'AR.ENTRIES]
      (if (NULL OLDDDESCRS)
          then ; No info stored yet, so include number as well, but no need to
            ; lengthen non-existent data. Also, flush old text info
            (\SETEOFPTR (AR.GET.SCRATCH.STREAM QFORMWINDOW)
                       0)
            (WINDOWPROP QFORMWINDOW 'AR.FIELD.DESCRPTIONS (CONS (create ARINDEXDESCR

```

```

ARINAME _ 'Number%:
ARIOFFKEYS _ 0)
DESCRS))
elseif (> NFIELDS 0)
then (for ENTRY in ENTRIES when (CDR ENTRY) do ; Only need to do this for entries that already have something
(NCONC ENTRY (to NFIELDS collect '?))
(replace (ARQUERYDATA ARQCOMPLETE) of ENTRY
with NIL))
(WINDOWPROP QFORMWINDOW 'AR.FIELD.DESCRPTIONS (NCONC OLDDDESCRS DESCRS))

```

(AR.KEYVALS.FROM.KEYLIST

```

[LAMBDA (KEYLIST) ; Edited 26-Feb-88 16:32 by bvm
;; Takes an AR index ENUMERATED.FIELD.KEYLIST and turns it into a list of values in order, such that the first element is the value for key 1.
(if (for VAL in (CDR KEYLIST) by (CDDR VAL) as I from 1 always (EQ I VAL))
then ;; Keys are in order, so it's easy to make the list. This is an optimization with the knowledge that all keys are currently stored this way.
(for KEY in KEYLIST by (CDDR KEY) collect KEY)
else (HELP "Enumerated keys out of order. RETURN to continue")
(LET ((KEYVALS (SORT (for TAIL on KEYLIST by (CDDR TAIL) collect (LIST (CADR TAIL)
(CAR TAIL)))
T)))
(for I from 1 while KEYVALS collect (AND (EQ I (CAAR KEYVALS))
(CADR (pop KEYVALS))
)
)

```

;; Printing summaries

(DEFINEQ

(AR.QFORM.SUMMARY

```

[LAMBDA (QFORMWINDOW) ; Edited 4-Aug-88 13:02 by bvm
;; Handles the "Hardcopy Summary" command -- sends summary straight to printer
(if (AR.QFORM.SUMMARIZE.CHECK QFORMWINDOW)
then (AR.PROMPT.PRINT QFORMWINDOW "Printing summary...")
(LET [(STREAM (AR.OPEN.IP.STREAM NIL (CONCAT AR.IDENTIFICATION.STRING " Summary")
(CL:UNWIND-PROTECT
(AR.PRINT.SUMMARY QFORMWINDOW STREAM)
(CLOSEF STREAM)
(AR.PROMPT.PRINT QFORMWINDOW "done."))]
)

```

(AR.QFORM.SUMMARY.TEXT

```

[LAMBDA (QFORMWINDOW) ; Edited 20-Jul-88 18:49 by bvm
;; Handle the "Text Summary" command--make a text file containing summary
(if (AR.QFORM.SUMMARIZE.CHECK QFORMWINDOW)
then (LET [(FILE (PROMPTFORWORD "File Name: " (WINDOWPROP QFORMWINDOW 'AR.SUMMARY.FILE.NAME)
NIL
(GETPROMPTWINDOW QFORMWINDOW)
NIL
'TTY]
(AR.PROMPT.CLEAR QFORMWINDOW)
(if FILE
then (WINDOWPROP QFORMWINDOW 'AR.SUMMARY.FILE.NAME FILE)
; Save specified name in case of abort, but later store the fully
; qualified name
(WINDOWPROP QFORMWINDOW 'AR.SUMMARY.FILE.NAME (PACKFILENAME.STRING 'VERSION NIL
'BODY
(SETQ FILE (
AR.MAKE.SUMMARY.TEXT.FILE
QFORMWINDOW FILE]
(AR.PROMPT.PRINT QFORMWINDOW T "Wrote " FILE)
else (AR.PROMPT.PRINT QFORMWINDOW " ... aborted"])]
)

```

(AR.MAKE.SUMMARY.FILE

```

[LAMBDA (QFORMWINDOW FILENAME FIELDS-TO-PRINT) ; Edited 29-Feb-88 19:42 by bvm
;; Write a summary file from the query in QFORMWINDOW to FILENAME. FILENAME can also be a stream on a file, such as from
;; OPENIPSTREAM.
(RESETLST
(LET* [(UPPER-CASE-FILE-NAMES*)
(STREAM (OR (STREAMP FILENAME)
(OPENSTREAM FILENAME 'OUTPUT 'NEW)
(RESETSAVE NIL (LIST 'CLOSE-AND-MAYBE-DELETE STREAM))
(AR.PROMPT.PRINT QFORMWINDOW T "Writing " (FULLNAME STREAM)
"...")
(AR.PRINT.SUMMARY QFORMWINDOW STREAM FIELDS-TO-PRINT)
(AR.PROMPT.PRINT QFORMWINDOW " done.")
(FULLNAME STREAM)))]
)

```

(AR.MAKE.SUMMARY.TEXT.FILE


```
[LAMBDA (QFORMWINDOW FILENAME) ; Edited 20-Jul-88 18:48 by bvm
  (LET ((*UPPER-CASE-FILE-NAMES* NIL))
    (CL:WITH-OPEN-FILE (S FILENAME :DIRECTION :OUTPUT :IF-EXISTS :NEW-VERSION)
      (AR.PRINT.SUMMARY QFORMWINDOW S)
      (FULLNAME S]))
```

(AR.QFORM.SUMMARY.TEDIT

```
[LAMBDA (QFORMWINDOW) ; Edited 4-Aug-88 13:02 by bvm
  (if (AR.QFORM.SUMMARIZE.CHECK QFORMWINDOW)
    then (LET* [(STREAM (OPENSTREAM '{NODIRCORE} 'BOTH))
                (FONT (FONTCREATE 'GACHA 8))
                (FIELDS (OR AR.TEDIT.FIELDS AR.SUMMARY.FIELDS))
                (WINDOW (CREATEW (GETBOXREGION (WIDTHIFWINDOW (TIMES (CHARWIDTH (CHARCODE X)
                                                                    FONT)
                                                                    (+ (for PAIR in FIELDS
                                                                    sum
                                                                    (+ 2 (CADR PAIR))))
                                                                    9)))
                                220)
                (CONCAT AR.IDENTIFICATION.STRING " Summary")
                (WINDOWPROP WINDOW 'ICONFN (FUNCTION TEXTICON))
                (AR.PROMPT.PRINT QFORMWINDOW "Creating summary...")
                (AR.PRINT.SUMMARY QFORMWINDOW STREAM FIELDS)
                (AR.PROMPT.PRINT QFORMWINDOW "done.")
                (TEDIT STREAM WINDOW NIL
                  `(LEAVETTY T FONT ,FONT PAGEFORMAT
                    , (TEDIT.SINGLE.PAGEFORMAT NIL NIL NIL NIL NIL 48 48 48 48 NIL NIL NIL NIL NIL
                    ' (LANDSCAPE? T]))
```

(AR.QFORM.SUMMARIZE.CHECK

```
[LAMBDA (QFORMWINDOW) ; Edited 26-Feb-88 19:37 by bvm
  ;; Returns true if there are ars in the browser window, else prints message and returns nil.
  (if (WINDOWPROP QFORMWINDOW 'AR.ENTRIES)
    else (AR.PROMPT.PRINT QFORMWINDOW "There are no ARs to summarize"
      NIL))
```

(AR.OPEN.IP.STREAM

```
[LAMBDA (FILE DOCUMENT.NAME) ; Edited 1-Aug-88 12:37 by bvm
  ;; Opens and returns an IP stream for printing an ar summary. FILE is the name given to OPENIMAGESTREAM, which is NIL for the default
  ;; printer. DOCUMENT.NAME is optional name to give the document (printer header page)
  (LET* [(UPPER-CASE-FILE-NAMES* NIL)
        (FONT (FONTCREATE '(TERMINAL 6)
                          NIL NIL NIL 'INTERPRESS))
        (REGION (CREATEREGION (fetch (REGION LEFT) of DEFAULTLANDPAGEREGION)
                              (- (fetch (REGION BOTTOM) of DEFAULTLANDPAGEREGION)
                                 (IQUOTIENT MICASPERINCH 2))
                              (fetch (REGION WIDTH) of DEFAULTLANDPAGEREGION)
                              (+ (fetch (REGION HEIGHT) of DEFAULTLANDPAGEREGION)
                                 MICASPERINCH)))
        (STREAM (OPENIMAGESTREAM FILE 'INTERPRESS
                              `(LANDSCAPE T REGION ,REGION HEADING
                                , (PROGN ; Crock. Make a heading consisting of enough spaces to get
                                        ; page number right justified. IP code spaces the page number
                                        ; an inch to the right of the heading
                                (ALLOCSTRING (IQUOTIENT (- (fetch (REGION WIDTH) of REGION)
                                                         (STRINGWIDTH "Page 999" FONT)
                                                         MICASPERINCH)
                                                         (CHARWIDTH (CHARCODE SPACE)
                                                         FONT))
                                             (CHARCODE SPACE)))
                                FONTS
                                (, FONT]
        (if DOCUMENT.NAME
          then (STREAMPROP STREAM 'PRINTOPTIONS (LIST* 'DOCUMENT.NAME DOCUMENT.NAME (STREAMPROP STREAM
                                                                                               'PRINTOPTIONS)
                                                                                               STREAM))
```

(AR.PRINT.PADDED

```
[LAMBDA (STR STREAM START MAXCHARS PRINTWIDTH) ; Edited 26-Feb-88 18:41 by bvm
  ;; Given a string or symbol to print, print characters from it, starting with char START, going for up to MAXCHARS. If PRINTWIDTH is supplied, it
  ;; must be at least MAXCHARS, and will result in padding the field with blanks on the right, as needed. If there were still chars left in the string at
  ;; the end, returns the offset of the next char to print, else NIL.
  (SETQ STREAM (\GETSTREAM STREAM 'OUTPUT))
  (LET (BASE STRLEN STROFF FATP CHARSLEFT START-OFFSET FATP EXCESSP)
    (if (LITATOM STR)
      then (SETQ BASE (ffetch (LITATOM PNAMEBASE) of STR))
          (SETQ STRLEN (ffetch (PNAMEBASE PNAMELENGTH) of BASE))
          (SETQ STROFF 1)
          (SETQ FATP (ffetch (LITATOM FATPNAMEP) of STR))
```



```

                (AND TEXTP (+ WIDTH 2)))
            (add COLUMN WIDTH 2)
            (LIST* (CL:POSITION NAME KNOWN.FIELDS :KEY 'CAR)
                WIDTH
                (CADDR TRIPLE])
(TERPRI STREAM)
(TERPRI STREAM)
(for DATA in ENTRIES bind ENTRY OVERFLOW.DATA FIELD.VALUES COLUMN LINE# NSPACES
do
    (PRINTOUT STREAM .I5 (fetch (ARQUERYDATA ARQ#) of DATA))
    (SETQ LINE# 0)
    (SETQ COLUMN 7)
    (SETQ NSPACES 2)
    (SETQ FIELD.VALUES (fetch (ARQUERYDATA ARQALLFIELDS) of DATA))
    [SETQ OVERFLOW.DATA
    (for D in DESCRS bind WIDTH VALUE collect D
    when (PROGN (SETQ WIDTH (CADR D))
                (SETQ VALUE (CL:NTH (CAR D)
                                    FIELD.VALUES))
        (PROG1 [if (NULL VALUE)
                then (add NSPACES WIDTH 2)
                    NIL
                else
                    ; Print the field, return T if there's more to print and it's not a field
                    ; restricted to one line
                    (if (NOT TEXTP)
                        then
                            ; Position to the correct column
                            (DSPXPOSITION (+ LMAR (TIMES SPACEWIDTH COLUMN))
                                STREAM))
                            (AND [AR.PRINT.SUMMARY.FIELD STREAM VALUE WIDTH 0 SCRATCH
                                (AND TEXTP (PROG1 NSPACES (SETQ NSPACES 2])
                                (NOT (CDDR D])
                            (add COLUMN WIDTH 2]
                    (TERPRI STREAM)
    (if AR.SUMMARY.MIN.LINES
        then
            ;; OVERFLOW.DATA is the set of descriptors that have more to do. Let's print some more lines
            (while OVERFLOW.DATA bind NEXTTOVERFLOW
            do (add LINE# 1)
                (SETQ COLUMN (SETQ NSPACES 7))
                (SETQ NEXTTOVERFLOW OVERFLOW.DATA)
                (for D in DESCRS bind WIDTH
                do (SETQ WIDTH (CADR D))
                    (if (NEQ D (CAR NEXTTOVERFLOW))
                        then
                            ; Not this field
                            (add NSPACES WIDTH 2)
                        else (if (NOT TEXTP)
                            then
                                ; Position to the correct column
                                (DSPXPOSITION (+ LMAR (TIMES SPACEWIDTH COLUMN))
                                    STREAM))
                                (if (NULL (AR.PRINT.SUMMARY.FIELD STREAM (CL:NTH (CAR D)
                                        FIELD.VALUES)
                                            WIDTH
                                            (TIMES LINE# WIDTH)
                                            SCRATCH
                                            (AND TEXTP NSPACES)))
                                    then (RPLACA NEXTTOVERFLOW NIL))
                                (if (NULL (SETQ NEXTTOVERFLOW (CDR NEXTTOVERFLOW)))
                                    then
                                        ; We're thru with this line
                                        (RETURN))
                                (SETQ NSPACES 2))
                            (add COLUMN WIDTH 2))
                (TERPRI STREAM)
            (SETQ OVERFLOW.DATA (DREMOVE NIL OVERFLOW.DATA))
        finally
            ; Ensure that we have printed enough lines
            (RPTQ (- AR.SUMMARY.MIN.LINES LINE# 1)
                (TERPRI STREAM])

```

(AR.PRINT.SUMMARY.FIELD

```

[LAMBDA (STREAM VALUE WIDTH START SCRATCH NSPACES)
    ; Edited 26-Feb-88 22:39 by bvm
    ;; Print specified VALUE in a field WIDTH wide, starting at offset START in the value (zero for the first line, width*#lines for later lines). SCRATCH
    ;; is the scratch stream where strings live. If we're printing to a plain text stream, NSPACES is the number of spaces required before we start
    ;; printing. Returns true if there is more to print after this.
    (if (AND NSPACES (NEQ NSPACES 0))
        then
            ; Need to get to starting column first
            (SPACES NSPACES STREAM))
    (if (LISTP VALUE)
        then
            ; String value = (ptr length) stored on scratch stream
            (SETFILEPTR SCRATCH (+ (CAR VALUE)
                                START))
            (LET ((LEN (- (CADR VALUE)
                        START)))
                (to (MIN LEN WIDTH) do (\OUTCHAR STREAM (BIN SCRATCH)))
                (if (> LEN WIDTH)
                    then
                        ; More to do...
                        T

```

```

else (if NSPACES
      then (SPACES (- WIDTH LEN) ; Value was shorter than field, so pad to end
            STREAM))
      NIL))
else (AR.PRINT.PADED VALUE STREAM (+ START 1)
      WIDTH
      (AND NSPACES WIDTH])
)

```

:: Evaluating AR queries

(DEFINEQ

(AR.QUERY

[LAMBDA (QFORMWINDOW QLIST SLIST) ; Edited 1-Aug-88 12:39 by bvm

:: Given a query window, and a query in the form of a list of items, run the query.

```

(WITH.AR.QUERY QFORMWINDOW
  (LET ((BROWSERWINDOW (WINDOWPROP QFORMWINDOW 'QFORM.ENTRY.WINDOW))
        TBROWSER QUERY.ENTRIES DISPLAY.FIELDS INDICES)
    (if [AND BROWSERWINDOW (SETQ TBROWSER (WINDOWPROP BROWSERWINDOW 'TABLEBROWSER])
        then ; Remove old items
        (TB.REPLACE.ITEMS TBROWSER NIL))
    (AR.PROMPT.PRINT QFORMWINDOW T "Searching...")
    (SETQ INDICES (AR.QUERY.EVAL QFORMWINDOW QLIST))
    (if (NULL INDICES)
        then (AR.PROMPT.PRINT QFORMWINDOW T "No matching ARs found.")
            (WINDOWPROP QFORMWINDOW 'AR.ENTRIES NIL))
    else ; Sort them and prepare to display them
    (AR.PROMPT.PRINT QFORMWINDOW T "Found " (LENGTH INDICES)
      " ARs. ")
    (WINDOWPROP QFORMWINDOW 'AR.FIELD.DESCRPTIONS NIL)
    [WINDOWPROP QFORMWINDOW 'AR.ENTRIES (SETQ INDICES (for x in INDICES
      collect (LIST X))
    (AR.AUGMENT.QUERY.FIELDS QFORMWINDOW
      (if BROWSERWINDOW
        then ; Will want to gather these fields for display
        (LET [(FIELDS (MAPCAR (SETQ DISPLAY.FIELDS AR.DISPLAY.FIELDS)
          (FUNCTION CAR))
          (if SLIST
            then ; Also need these additional fields to sort by
                (APPEND FIELDS (CL:SET-DIFFERENCE SLIST FIELDS))
            else FIELDS))
          else SLIST))
        (if (AR.QUERY.SMALLP QFORMWINDOW BROWSERWINDOW INDICES)
          then ; Small enough to fetch everything at once
          (AR.ENSURE.QUERY.DATA QFORMWINDOW T))
        (if SLIST
          then (WINDOWPROP QFORMWINDOW 'AR.ENTRIES (SETQ INDICES (AR.SORT.BY QFORMWINDOW INDICES
            SLIST])
            (WINDOWPROP QFORMWINDOW 'AR.ENTRY.ALIST.SLIST SLIST)
            (WINDOWPROP QFORMWINDOW 'AR.DISPLAY.FIELDS DISPLAY.FIELDS)
            (if BROWSERWINDOW
              then ; Install these guys in a TableBrowser
              (SETQ QUERY.ENTRIES (for ENTRY in INDICES collect (create TABLEITEM
                TIDATA _ ENTRY)))
              (if TBROWSER
                then (TB.REPLACE.ITEMS TBROWSER QUERY.ENTRIES)
                else (TB.MAKE.BROWSER QUERY.ENTRIES BROWSERWINDOW (LIST 'PRINTFN
                  (FUNCTION
                    AR.BROWSER.PRINTFN]
            (AR.PROMPT.PRINT QFORMWINDOW " done. "))
            (WINDOWPROP QFORMWINDOW 'AR.ENTRY.ALIST.QLIST QLIST]))

```

(AR.QUERY.SMALLP

[LAMBDA (QFORMWINDOW BROWSERWINDOW ENTRIES) ; Edited 26-Jul-88 11:20 by bvm

:: True if query is small enough to be worth fetching all its data en masse.

:: Current def: true if all entries will fit in window (since then we'll have to fetch all anyway).

```

(<= (LENGTH ENTRIES)
  (IQUOTIENT (WINDOWPROP BROWSERWINDOW 'HEIGHT)
    (FONTPROP BROWSERWINDOW 'HEIGHT]))

```

(AR.QUERY.EVAL

[LAMBDA (QFORMWINDOW QLIST ANDINDEXES NEGFLG) ; Edited 15-Mar-88 20:23 by bvm

:: Given a query spec in QLIST, evaluate it and return a list of indices that meet the criteria. If ANDINDEXES is non-NIL, must AND this query with them. If NEGFLG is true, want to evaluate (NOT QLIST).

```

(COND
  ((NLISTP QLIST) ; The spec wasn't a list, so it isn't valid.
  (AR.BAD.QUERY QFORMWINDOW QLIST))
  (T (SELECTQ (CAR QLIST)

```

```
(AND (COND
      (NEGFLG (AR.QUERY.NAND QFORMWINDOW (CDR QLIST)
                                ANDINDEXES))
      (T (AR.QUERY.AND QFORMWINDOW (CDR QLIST)
                                ANDINDEXES))))
(OR (if NEGFLG
      then ; (NOT (OR x y)) = (AND (NOT x) (NOT y))
      (AR.QUERY.AND QFORMWINDOW (CDR QLIST)
                    ANDINDEXES T)
     else (AR.QUERY.OR QFORMWINDOW (CDR QLIST)
                    ANDINDEXES)))
(NOT (if (OR (NULL (CDR QLIST))
            (CDDR QLIST))
        then ; NOT takes exactly one clause
        (AR.BAD.QUERY QFORMWINDOW QLIST)
       else (AR.QUERY.EVAL QFORMWINDOW (CADR QLIST)
                    ANDINDEXES
                    (NOT NEGFLG))))
(SELECTQ (CADR QLIST)
  (HAS (AR.QUERY.HAS QFORMWINDOW (CAR QLIST)
                    (CADDR QLIST)
                    ANDINDEXES NEGFLG)) ; String search
  (IS (AR.QUERY.IS QFORMWINDOW (CAR QLIST)
                   (CADDR QLIST)
                   ANDINDEXES NEGFLG)) ; Enumeration search
  ((> >= < <= = ~=)
   (AR.QUERY.COMPARE QFORMWINDOW QLIST ANDINDEXES NEGFLG))
  (AR.BAD.QUERY QFORMWINDOW QLIST))
```

(AR.BAD.QUERY

```
[LAMBDA (QFORMWINDOW ITEM) ; Edited 25-Feb-88 11:57 by bvm
  (AR.PROMPT.PRINT QFORMWINDOW T "Bad Query Spec: " ITEM)
  (ERROR!)]
```

(AR.QUERY.AND

```
[LAMBDA (QFORMWINDOW CLAUSES ANDINDEXES NEGFLG RECURSIVE-P) ; Edited 21-Jul-88 18:48 by bvm
  ;; Compute the intersection of ANDINDEXES with the evaluation of each of CLAUSES. ANDINDEXES of NIL means T to get this going. NEGFLG
  ;; means take negation of each clause
  (if (CDR CLAUSES)
      then (SETQ CLAUSES (AR.QUERY.SORT.CLAUSES QFORMWINDOW CLAUSES NEGFLG)))
  (if (NULL ANDINDEXES)
      then (SETQ ANDINDEXES (AR.QUERY.EVAL QFORMWINDOW (pop CLAUSES)
                                           NIL NEGFLG)))
  (for C in CLAUSES while ANDINDEXES do (if (NOT RECURSIVE-P)
      then ; Give progress report at top level query
      (AR.PROMPT.PRINT QFORMWINDOW "(" (LENGTH ANDINDEXES)
                        ") ")
      (SETQ ANDINDEXES (AR.QUERY.EVAL QFORMWINDOW C ANDINDEXES NEGFLG)))
  finally (RETURN ANDINDEXES])
```

(AR.QUERY.NAND

```
[LAMBDA (QFORMWINDOW CLAUSES ANDINDEXES) ; Edited 21-Jul-88 15:33 by bvm
  (AR.QUERY.COMBINE.RESULT (AR.QUERY.AND QFORMWINDOW CLAUSES NIL NIL T)
    ANDINDEXES T)]
```

(AR.QUERY.SORT.CLAUSES

```
[LAMBDA (QFORMWINDOW CLAUSES NEGFLG) ; Edited 17-Jan-89 19:21 by SYBALSKY
  ;; Sort CLAUSES into a preferred order for an AND query. If NEGFLG is true, we'll actually be querying the negation of each clause.
  (LET
    ((SORT.ORDER (AR.QUERY.SORT.ORDER QFORMWINDOW))
     [if (for C in CLAUSES thereis (SELECTQ (CAR C)
      ((AND NOT
        T)
        NIL))
      then
      ;; First canonicalize any funny clauses
      (LET*
        ((HEAD (CONS NIL CLAUSES))
         (PREV HEAD)
         (TAIL (CDR PREV))
         C NEWTAIL)
        [while TAIL
         do (if (SETQ NEWTAIL (SELECTQ (CAR (SETQ C (CAR TAIL)))
          (AND ; Spread any top-level AND
            (NCONC (CDR C)
                  (CDR TAIL)))
            (NOT (if (EQ (CAR (LISTP (CADR C)))
                      'OR
```

```

then ; (NOT (OR --)) => (AND (NOT ..) --)
(NCONC [for CL in (CDADR C)
collect `(NOT ,CL]
(CDR TAIL)))]
NIL))
then (RPLACD PREV (SETQ TAIL NEWTAIL))
else (SETQ TAIL (CDR (SETQ PREV TAIL)
(SETQ CLAUSES (CDR HEAD]
;; Assign each clause a value, sort the list numerically, then pull the clauses back out.
(MAPCAR (CL:STABLE-SORT (for C in CLAUSES collect (CONS (AR.QUERY.SORT.VALUE C SORT.ORDER NEGFLG)
C))
(FUNCTION <)
:KEY
(FUNCTION CAR))
(FUNCTION CDR]))

```

(AR.QUERY.SORT.ORDER

[LAMBDA (QFORMWINDOW)

; Edited 22-Jul-88 10:57 by bvm

;; Fetch or compute the sort order for this query window, which is heuristically based on database characteristics. Value is (enumerated . strings),
 ;; where each component is a list of the attributes in order of "best to query first", preceded by the length of the list. We believe that it is best to
 ;; query enumerated attributes before string attributes (easier search). Within enumerated attributes it is best to search first on those that divide the
 ;; space more thoroughly (which we heuristically determine to be inversely related to the number of possible values it takes on). Within string
 ;; attributes, it is best to search first for those that take up less space on the file, since that will require fewer file accesses.

```

(OR (WINDOWPROP QFORMWINDOW 'AR.QUERY.SORT.ORDER)
(LET (ENUMERATED STRINGS ORDER KEYS)
[for SPEC in AR.INDEX.FIELD.SPECS do
; Spec = (field . plist)
(if (SETQ KEYS (LISTGET (CDR SPEC)
'ENUMERATED.FIELD.KEYLIST))
then ; Judge enumerated fields by the number of possible values
(push ENUMERATED (LIST (LENGTH KEYS)
(CAR SPEC)))
else ; Judge string fields by how much space they take in the file
(push STRINGS (LIST (- (LISTGET (CDR SPEC)
'FIELD.END.PTR)
(LISTGET (CDR SPEC)
'FIELD.BEGIN.PTR))
(CAR SPEC)

```

;; Enumerated fields is better to have large values. For strings, better to have small values.

```

(SETQ ORDER (for PAIRS in (LIST (REVERSE (SORT ENUMERATED T))
(SORT STRINGS T))
bind (I _ 0)
ORIGI LASTVALUE THISORDER
collect ; Process Enumerated, then Strings
(SETQ ORIGI I)
(SETQ LASTVALUE NIL)
[SETQ THISORDER (for PAIR in PAIRS
join (LIST (CADR PAIR)
(if (EQ LASTVALUE (SETQ LASTVALUE (CAR PAIR)))
then
; Same priority
I
else (add I 1]
; Finally, tack on the front a number that you can subtract any
; value from in order to negate the sense of the list
(CONS (+ I ORIGI 1)
THISORDER))
(RPLACD ORDER (CADR ORDER))
(WINDOWPROP QFORMWINDOW 'AR.QUERY.SORT.ORDER ORDER)
ORDER])

```

(AR.QUERY.SORT.VALUE

[LAMBDA (CLAUSE SORT.ORDER NEGFLG)

; Edited 25-Jul-88 12:02 by bvm

;; Assign a value to CLAUSE. Low values mean search for me sooner.

```

(while (EQ (CAR CLAUSE)
'NOT)
do (SETQ NEGFLG (NOT NEGFLG))
(SETQ CLAUSE (CADR CLAUSE)))
(SELECTQ (CAR CLAUSE)
(AND ; Take the minimum of the clauses
(for C in (CDR CLAUSE) bind (RESULT _ 1000) do (SETQ RESULT (MIN RESULT (AR.QUERY.SORT.VALUE C
SORT.ORDER NEGFLG)))
finally (RETURN RESULT)))
(OR ; Take the maximum, since we'll have to query ALL the clauses
; and take the union
(for C in (CDR CLAUSE) bind (RESULT _ -1) do (SETQ RESULT (MAX RESULT (AR.QUERY.SORT.VALUE C
SORT.ORDER NEGFLG)))
finally (RETURN RESULT)))
(LET* [(OP (CADR CLAUSE))
(ORDER (if (EQ OP 'HAS)
then ; String search
(CDR SORT.ORDER)

```

```

else ; Enumerated or maybe a weird one
  (if (EQ OP '~=)
      then (SETQ NEGFLG (NOT NEGFLG))
          (CAR SORT.ORDER))
  (V (OR (LISTGET (CDR ORDER)
                (CAR CLAUSE))
        (CAR ORDER)
        (if (AND NEGFLG (NEQ OP 'HAS))
            then ; Reverse order within this list. This is only vaguely right, and does nothing to account for relational operators (<= etc).
                ; Don't reverse order for string search ops, since their order is a function of how much file there is to search, which
                ; doesn't change when negated.
                (- (CAR ORDER)
                  V)
            else V]))

```

(AR.QUERY.OR

```

[LAMBDA (QFORMWINDOW CLAUSES ANDINDEXES) ; Edited 10-Mar-88 18:35 by bvm
  ;; Take the OR of clauses, ANDed with ANDINDEXES (if non-nil). Since (AND X (OR Y Z)) = (OR (AND X Y) (AND X Z)), we can just pass
  ;; ANDINDEXES along to the subqueries
  (for C in CLAUSES bind A B
    do (SETQ B (AR.QUERY.EVAL QFORMWINDOW C ANDINDEXES))
      [SETQ A (NCONC (while (AND B A) collect (if (< (CAR B)
                                                    (CAR A))
                                                then ; B < A, so take B
                                                (pop B)
                                                else ; A <= B, so take at least A
                                                (if (NOT (< (CAR A)
                                                            (CAR B)))
                                                    then ; A = B, so pop from both
                                                    (pop B))
                                                (pop A)))
                    (PROGN (OR B A) ; Plus whichever, if either, is left over
                           (RETURN A])
    finally (RETURN A])

```

(AR.QUERY.COMBINE.RESULT

```

[LAMBDA (INDEXES ANDINDEXES NEGFLG) ; Edited 25-Jul-88 15:33 by bvm
  ;; Used by query handlers that don't handle ANDINDEXES and NEGFLG as part of their operation already. If NEGFLG is true, complements
  ;; INDEXES. Then if ANDINDEXES is given, intersects the result with ANDINDEXES.
  (if ANDINDEXES
      then ; Intersect INDEXES and ANDINDEXES by collecting everything in ANDINDEXES that is also in (or NOT in if NEGFLG) INDEXES,
          ; taking advantage of the fact that both lists are in order.
          (for I in ANDINDEXES collect I
            unless (EQ [do (if (NULL INDEXES)
                              then ; No more in INDEXES
                              (RETURN NIL)
                              elseif (> I (CAR INDEXES))
                              then ; We've passed by some elements of INDEXES, so throw them
                                      ; out
                                      (SETQ INDEXES (CDR INDEXES))
                              else ; At this point next element of INDEXES is at least I
                                      (RETURN (if (EQ I (CAR INDEXES))
                                                  then ; It's equal to I, so signal true (change EQ to >= if indexes can be
                                                          ; as big as 2^16)
                                                  (SETQ INDEXES (CDR INDEXES))
                                                  T]
                                                NEGFLG))
            elseif NEGFLG
              then ; Compute the complement of INDEXES
                (for I from 0 to AR.MAX.INDEX when (COND
                  ((OR (NULL INDEXES)
                      (< I (CAR INDEXES)))
                   ; Haven't hit the next one in INDEXES.
                   T)
                  (T ; Omit this one, and pop it off the list. Since INDEXES is dense
                    ; and sorted, it must be the case that I = (car indexes)
                    (OR (EQ I (CAR INDEXES))
                        (HELP))
                    (pop INDEXES)
                    NIL))
                collect I)
            else INDEXES])
  )

```

(DEFINEQ

(AR.QUERY.IS

```

[LAMBDA (QFORMWINDOW FIELD.NAME VALUE ANDINDEXES NEGFLG) ; Edited 16-Mar-88 12:25 by bvm
  ;; Equality search for enumerated fields. If ANDINDEXES is supplied, result is AND of them and this search. NEGFLG means search for those
  ;; whose field is NOT this value.

```

```
(LET* [(FIELD.SPEC (CDR (ASSOC FIELD.NAME AR.INDEX.FIELD.SPECS)))
      (FIELD.KEYLIST (LISTGET FIELD.SPEC 'ENUMERATED.FIELD.KEYLIST)
      (if FIELD.KEYLIST
        then
          (AR.QUERY.IS.EXACTLY QFORMWINDOW FIELD.NAME (if (NULL VALUE)
            then 0
            elseif (LISTGET FIELD.KEYLIST
              (if (LITATOM VALUE)
                then VALUE
                else (MKATOM VALUE)))
            else (AR.PROMPT.PRINT QFORMWINDOW T "Unknown
              value " VALUE " for field: "
              FIELD.NAME)
              (ERROR!))
              (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR)
              ANDINDEXES NEGFLG)
        elseif (NOT (MEMB FIELD.NAME AR.INDEX.FIELD.LIST))
          then (AR.PROMPT.PRINT QFORMWINDOW T "Unknown field name: " FIELD.NAME)
              (ERROR!)
        elseif (OR (NULL VALUE)
          (EQ (NCHARS VALUE)
            0))
          then
            ; We're willing to search for empty string fields
            (AR.QUERY.IS.EMPTY QFORMWINDOW FIELD.NAME ANDINDEXES NEGFLG)
        else (AR.PROMPT.PRINT QFORMWINDOW T "Can't use IS on non-enumerated field " FIELD.NAME " -- will use
          HAS")
          (AR.QUERY.HAS QFORMWINDOW FIELD.NAME VALUE ANDINDEXES NEGFLG]))
```

(AR.QUERY.IS.EXACTLY

```
[LAMBDA (QFORMWINDOW FIELD.NAME SEARCH.KEY BEGIN ANDINDEXES NEGFLG)
  ; Edited 25-Jul-88 15:33 by bvm
  ;; Searches for ARs whose enumerated FIELD.NAME is exactly SEARCH.KEY, a numeric value we have already figured out. The values for this
  ;; field are all stored in consecutive bytes on the file. Just gobble up the bytes, collecting index when byte matches search key
  (AR.PROMPT.PRINT QFORMWINDOW FIELD.NAME " ")
  (if ANDINDEXES
    then
      ; Only look at the specified AR's.
      (LET ((LASTINDEX (CAR ANDINDEXES)))
        (SETFILEPTR AR.INDEX.FILE (+ BEGIN LASTINDEX))
        (for INDEX in ANDINDEXES when (PROGN (\INCFILEPTR AR.INDEX.FILE (- INDEX LASTINDEX))
          (SETQ LASTINDEX (ADD1 INDEX))
          (NEQ (EQ (BIN AR.INDEX.FILE)
            SEARCH.KEY)
            NEGFLG))
          collect INDEX))
    else (SETFILEPTR AR.INDEX.FILE BEGIN)
      (for INDEX from 0 to AR.MAX.INDEX when (NEQ (EQ (BIN AR.INDEX.FILE)
        SEARCH.KEY)
        NEGFLG)
        collect INDEX])
```

(AR.QUERY.COMPARE.ENUMERATED

```
[LAMBDA (QFORMWINDOW CLAUSE ANDINDEXES NEGFLG FIELD.KEYLIST BEGIN)
  ; Edited 25-Jul-88 15:33 by bvm
  ;; Perform a numeric comparison on an enumerated field. CLAUSE is the query, in form (field.name op value [op value]). FIELD.KEYLIST is the
  ;; set of keys and BEGIN is where the field values start in the index file.
  (DESTRUCTURING-BIND (OP.HI HI.NUM OP.LO LO.NUM NEGFLG)
    [AR.QUERY.COMPARE.PARSE QFORMWINDOW CLAUSE NEGFLG (FUNCTION (LAMBDA (VALUE QFORMWINDOW)
      ;; Turn an enumerated field value into a search
      ;; key
      (if (NULL VALUE)
        then 0
        else
          (LISTGET FIELD.KEYLIST
            (if (LITATOM VALUE)
              then VALUE
              else (MKATOM VALUE))
            ))
      ))
    ;; At this point, OP.HI is one of >, >= or =, HI.NUM is corresponding search key. Optional Lower bound is in OP.LO & LO.NUM. This is
    ;; backwards from the usual comparison parse: from the user's point of view, > really means <, since the "largest" key is 0 (nil), largest
    ;; non-nil key is 1, etc.
    ;; The values for this field are all stored in consecutive bytes on the file. Just gobble up the bytes, collecting index when byte compares
    ;; properly against search key
    (if (EQ OP.HI '=)
      then
        ; We already have someone to do this search
        (AR.QUERY.IS.EXACTLY QFORMWINDOW (CAR CLAUSE)
          HI.NUM BEGIN ANDINDEXES NEGFLG)
      else (LET (KEY LASTINDEX)
        (AR.PROMPT.PRINT QFORMWINDOW (CAR CLAUSE)
          " ")
        (if (EQ OP.HI '>')
          then
            ; Exclude the bound
```



```

      (add HI.NUM -1))
  (if (EQ OP.LO '<')
      then
      (add LO.NUM 1))
  (if ANDINDEXES
      then
      [SETFILEPTR AR.INDEX.FILE (+ BEGIN (SETQ LASTINDEX (CAR ANDINDEXES)
      (for INDEX in ANDINDEXES when (PROGN (\INCFILEPTR AR.INDEX.FILE (- INDEX LASTINDEX
      ))
      (SETQ LASTINDEX (ADD1 INDEX))
      (NEQ (AND (<= (SETQ KEY (BIN AR.INDEX.FILE))
      HI.NUM)
      (OR (NULL LO.NUM)
      (>= KEY LO.NUM)))
      NEGFLG))
      collect INDEX)
  else (SETFILEPTR AR.INDEX.FILE BEGIN)
      (for INDEX from 0 to AR.MAX.INDEX when (NEQ (AND (<= (SETQ KEY (BIN AR.INDEX.FILE))
      HI.NUM)
      (OR (NULL LO.NUM)
      (>= KEY LO.NUM)))
      NEGFLG)
      collect INDEX])

```

(AR.QUERY.IS.EMPTY

```

[LAMBDA (QFORMWINDOW FIELD.NAME ANDINDEXES NEGFLG) ; Edited 25-Jul-88 15:34 by bvm
;; Query on a non-enumerated field for values that are null
;; Algorithm: Walk thru the fixed-size entries for each AR, and collect the index when the next guy's field ptr is the same as this one's, i.e., the text
;; length is 0.
(AR.PROMPT.PRINT QFORMWINDOW FIELD.NAME " ")
(LET* ((FIELD.SPEC (CDR (ASSOC FIELD.NAME AR.INDEX.FIELD.SPECS)))
      (OFFSET (LISTGET FIELD.SPEC 'FIELD.OFFSET))
      [N+1.VALUE (PROGN
      ;; Since the length of an entry is computed by subtracting its pointer from the next entry's pointer, you can only
      ;; compute the final length by looking at the length of the whole region. Sure would be nice if the index always
      ;; had a last ar + 1 entry, instead of using FIELD.END.PTR
      (- (LISTGET FIELD.SPEC 'FIELD.END.PTR)
      (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR)
      (INCREMENT (- AR.INDEX.ENTRY.SIZE AR.BYTES.PER.PTR))
      (STREAM AR.INDEX.FILE)
      (MAX.INDEX AR.MAX.INDEX))
      (if ANDINDEXES
          then
          (for INDEX in ANDINDEXES bind (LASTINDEX _ (PROGN
          ; Only look at these entries
          ; Initially position stream as if we had read the first one.
          (SETFILEPTR STREAM (+ (
          AR.ENTRY.PTR.FROM.INDEX
          (CAR ANDINDEXES)
          OFFSET)
          AR.BYTES.PER.PTR))
          (CAR ANDINDEXES)))
          collect INDEX when (PROGN (\INCFILEPTR STREAM (- (TIMES (- INDEX LASTINDEX)
          AR.INDEX.ENTRY.SIZE)
          AR.BYTES.PER.PTR))
          (EQ (< (\DWIN STREAM)
          (if (EQ INDEX MAX.INDEX)
              then N+1.VALUE
              else ; Read the next value, and note LASTINDEX belonging here
              (SETQ LASTINDEX (ADD1 INDEX))
              (\INCFILEPTR STREAM INCREMENT)
              (\DWIN STREAM)))
          NEGFLG)))
          else
          ; Search all ARs. For this, we optimize by reading each pointer
          ; only once.
          (for INDEX from 0 bind DONE (LASTPTR _ (PROGN
          ; Initialize loop by reading the address (offset) of the first AR's
          ; field value
          (SETFILEPTR STREAM (AR.ENTRY.PTR.FROM.INDEX 0
          OFFSET))
          (\DWIN STREAM)))
          until DONE when (PROGN
          ;; Bump the file pointer to the place where the address (actually, offset) of the next ar's value is
          ;; stored. If pointers are the same, value of last AR's field must be null. Since the pointers are
          ;; monotonic, we can use < instead of the potentially slower = -- if this < next, then entry is non-null.
          (\INCFILEPTR STREAM INCREMENT)
          (EQ [< LASTPTR (SETQ LASTPTR (if (EQ INDEX MAX.INDEX)
          then (SETQ DONE T)
          ; ptr for n+1'st entry computed artificially
          N+1.VALUE
          else (\DWIN STREAM)
          NEGFLG))
          collect INDEX])
)

```

(DEFINEQ

(AR.QUERY.HAS

[LAMBDA (QFORMWINDOW FIELD.NAME SEARCH.STRING ANDINDEXES NEGFLG)

; Edited 25-Jul-88 15:34 by bvm

;; Find ARs containing SEARCH.STRING in their FIELD.NAME. If ANDINDEXES is given, search only those ars. NEGFLG=T means search for ARs NOT containing the string.

(LET*

[(FIELD.SPEC (CDR (ASSOC FIELD.NAME AR.INDEX.FIELD.SPECS)))
(FIELD.OFFSET (LISTGET FIELD.SPEC 'FIELD.OFFSET))

(if (NULL FIELD.OFFSET)

then ; Not a variable field

(if (NOT (MEMB FIELD.NAME AR.INDEX.FIELD.LIST))

then (AR.PROMPT.PRINT QFORMWINDOW T "Unknown field name: " FIELD.NAME)

(ERROR!)

else (AR.PROMPT.PRINT QFORMWINDOW T "Can't use HAS on enumerated field " FIELD.NAME " -- will use IS")

(AR.QUERY.IS QFORMWINDOW FIELD.NAME SEARCH.STRING ANDINDEXES NEGFLG))

elseif (OR (NULL SEARCH.STRING)

(EQ 0 (NCHARS SEARCH.STRING)))

then ; Search for empty field

(AR.QUERY.IS.EMPTY QFORMWINDOW FIELD.NAME ANDINDEXES NEGFLG)

else ; The text of all values of this field for all ARs is stored consecutively. Search that region of the index for desired string, then translate those file pointers into indices. If ANDINDEXES is given, can restrict search to a narrower range

(AR.PROMPT.PRINT QFORMWINDOW FIELD.NAME " ")

(PROG* ((PATLENGTH (NCHARS SEARCH.STRING))

(BEGIN (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR))

(HI.PTR (- (LISTGET FIELD.SPEC 'FIELD.END.PTR)

BEGIN))

(HI.INDEX (ADD1 AR.MAX.INDEX))

LO.PTR LO.INDEX)

(if ANDINDEXES

then [if (AR.SPARSE.QUERY PTR HI.PTR ANDINDEXES)

then

; The text to search comes out to less than one per 2 pages, so
; it's likely to be faster to search the ARs one at a time.

(RETURN (for INDEX in ANDINDEXES as SHAPE in (AR.COLLECT.SHAPES ANDINDEXES FIELD.OFFSET HI.PTR)

bind START collect INDEX

when (NEQ (AND (NEQ (CADR SHAPE)

0)

(FILEPOS SEARCH.STRING AR.INDEX.FILE

(SETQ START (+ BEGIN (CAR SHAPE)))

(+ START (- (CADR SHAPE)

PATLENGTH))

NIL T UPPERCASEARRAY))

NEGFLG]

(SETQ LO.PTR (AR.ENTRY.VALUE.FROM.INDEX (SETQ LO.INDEX (CAR ANDINDEXES) FIELD.OFFSET))

(if [NEQ HI.INDEX (SETQ HI.INDEX (ADD1 (CAR (LAST ANDINDEXES)

then (SETQ HI.PTR (AR.ENTRY.VALUE.FROM.INDEX HI.INDEX FIELD.OFFSET))])

else ; Nothing to go on, search everything

(SETQ LO.PTR (SETQ LO.INDEX 0))

(SETFILEPTR AR.INDEX.FILE (+ BEGIN LO.PTR))

(RETURN (AR.QUERY.COMBINE.RESULT (AR.INDICES.FROM.FILEPTRS

(bind (LAST.POS _ (- (+ HI.PTR BEGIN) PATLENGTH))

PTR

while (SETQ PTR

(FILEPOS SEARCH.STRING AR.INDEX.FILE NIL

LAST.POS NIL T UPPERCASEARRAY))

collect ; remember that these pointers are to the filepos AFTER the last
; char of the match

(- PTR BEGIN))

LO.INDEX HI.INDEX FIELD.OFFSET LO.PTR HI.PTR PATLENGTH)

ANDINDEXES NEGFLG])

(AR.COLLECT.SHAPES

[LAMBDA (INDEXES OFFSET TOTALSIZE)

; Edited 25-Jul-88 15:34 by bvm

;; For each of INDEXES, collect the offset and length of its OFFSET entry. TOTALSIZE is the offset of the fictional last+1 entry.

(LET* ((STREAM AR.INDEX.FILE)

(MAX.INDEX AR.MAX.INDEX)

(INCREMENT (- AR.INDEX.ENTRY.SIZE AR.BYTES.PER.PTR))

(LASTINDEX (PROGN

(SETFILEPTR STREAM (+ (AR.ENTRY.PTR.FROM.INDEX (CAR INDEXES)

OFFSET)

AR.BYTES.PER.PTR))

(CAR INDEXES)))

START)

(for INDEX in INDEXES collect (\INCFILEPTR STREAM (- (TIMES (- INDEX LASTINDEX)

AR.INDEX.ENTRY.SIZE)

AR.BYTES.PER.PTR))

(LIST (SETQ START (\DWIN STREAM)

(- (if (EQ INDEX MAX.INDEX)

```

                then TOTALSIZE
            else ; Read the next value, and note LASTINDEX belonging here
                (SETQ LASTINDEX (ADD1 INDEX))
                (\INCFILEPTR STREAM INCREMENT)
                (\DWIN STREAM))
        START])

```

(AR.COLLECT.SIZES

```

[LAMBDA (LO.INDEX HI.INDEX OFFSET MAX.INDEX TOTALSIZE) ; Edited 21-Mar-88 17:58 by bvm
;; Collect just the lengths of the OFFSET'th field of ars from LO.INDEX to HI.INDEX
(for INDEX from LO.INDEX to HI.INDEX bind (INCREMENT _ (- AR.INDEX.ENTRY.SIZE AR.BYTES.PER.PTR))
(LASTPTR _ (PROGN ; Initialize loop by reading the address (offset) of the first AR's
; field value
(AR.ENTRY.VALUE.FROM.INDEX LO.INDEX OFFSET)))
(STREAM _ AR.INDEX.FILE)

collect
;; Bump the file pointer to the place where the address (actually, offset) of the next ar's value is stored.
(\INCFILEPTR STREAM INCREMENT)
(- (- LASTPTR (SETQ LASTPTR (if (EQ INDEX MAX.INDEX)
then TOTALSIZE ; ptr for n+1'st entry computed artificially
else (\DWIN STREAM]))

```

(AR.SPARSE.QUERY

```

[LAMBDA (DATALENGTH ANDINDEXES) ; Edited 15-Mar-88 12:53 by bvm
;; Return true if we believe that a HAS search in a space of DATALENGTH bytes confined to the ars ANDINDEXES is likely to be faster by
;; searching individual ARs than by searching the whole space.
;; Current heuristic: if there is on average fewer than one AR (of ANDINDEXES) per data page, we'll save file accesses by searching specially.
;; This is fairly conservative--there are many searches in which we would win even if the average is bigger than 1, just because those references
;; may clump.
(> (FOLDLO DATALENGTH BYTESPERPAGE)
(LENGTH ANDINDEXES])

```

(AR.INDICES.FROM.FILEPTRS

```

[LAMBDA (FILEPTRS LO.INDEX HI.INDEX FIELD.OFFSET LO.PTR HI.PTR PATLENGTH) ; Edited 17-Mar-88 12:18 by bvm
;; Perform binary search on the index to compute the index pointers for fields returned from FFILEPOS. FILEPTRS is a list of pointers to the
;; character after a successful search. They are known to correspond to indices in [lo.index, hi.index). Those indices correspond to file pointers
;; LO.PTR and HI.PTR. PATLENGTH is the length of the pattern, which we need in order to determine whether a candidate file pointer is good, or
;; overlaps two ars.
(PROG ((NUMARS (- HI.INDEX LO.INDEX))
MID.INDEX MID.PTR NEXT.PTR BYTES.PER.INDEX)
(if (NULL FILEPTRS)
then (RETURN NIL))
(if (EQ NUMARS 0)
then (HELP "HI=LO and still have fileptrs to find.)))
[SETQ MID.INDEX (if (OR (CDR FILEPTRS)
(EQ (SETQ BYTES.PER.INDEX (IQUOTIENT (- HI.PTR LO.PTR)
NUMARS))
0))
then ; Pick the midpoint of the range, and then divide FILEPTRS into
; those that fall below it, those that match it, and those that fall
; after it.
(+ LO.INDEX (IQUOTIENT NUMARS 2))
else ; Down to searching for just one element, so try to get closer
; than just stabbing at the midpoint
(+ LO.INDEX (IMIN (IQUOTIENT (- (CAR FILEPTRS)
LO.PTR)
BYTES.PER.INDEX)
(SUB1 NUMARS)
(if (EQ (- HI.INDEX MID.INDEX)
1)
then ; Next = HI
(SETQ NEXT.PTR HI.PTR))
[if (EQ MID.INDEX LO.INDEX)
then (SETQ MID.PTR LO.PTR)
else (SETQ MID.PTR (AR.ENTRY.VALUE.FROM.INDEX MID.INDEX FIELD.OFFSET))
; Fileptr corresponding to MID.INDEX. This is the largest value a
; pointer can take and belong to an entry below MID.INDEX
(if (NOT NEXT.PTR)
then ; Find start of next entry. Pointers in (mid, next] belong to
; MID.INDEX
(SETQ NEXT.PTR (AR.ENTRY.VALUE.NEXT]
(RETURN
(for (TAIL _ FILEPTRS) bind PREV
do ; Search for the midpoint of the list, i.e., the place where all the pointers preceding it are before MID.PTR
(if (NULL TAIL)
then ; Everything comes before MID.INDEX
(RETURN (AR.INDICES.FROM.FILEPTRS FILEPTRS LO.INDEX MID.INDEX FIELD.OFFSET LO.PTR

```

```

                                MID.PTR PATLENGTH))
elseif (> (CAR TAIL)
         MID.PTR)
  then
    [RETURN (NCONC (if (NULL PREV)
                      then
                        NIL
                      else (RPLACD PREV NIL)
                      )
             ; Snip off prefix
             (AR.INDICES.FROM.FILEPTRS FILEPTRS LO.INDEX MID.INDEX FIELD.OFFSET
             LO.PTR MID.PTR PATLENGTH))
    (if [<= (CAR TAIL)
        (OR NEXT.PTR (SETQ NEXT.PTR (AR.ENTRY.VALUE.FROM.INDEX
                                     (ADD1 MID.INDEX)
                                     FIELD.OFFSET]
        then
          ; One or more of these pointers falls in the MID.INDEX range.
          ; Get rid of all of them.
          (AND (when (>= (- (pop TAIL)
                           MID.PTR)
                           PATLENGTH)
                do
                  ; The entire pattern is at or beyond MID.PTR, so it's a legitimate
                  ; match
                  (SETQ $$VAL T)
                repeatwhile (AND TAIL (<= (CAR TAIL)
                                           NEXT.PTR)))
          (LIST MID.INDEX)))
    (AND TAIL (AR.INDICES.FROM.FILEPTRS TAIL (ADD1 MID.INDEX)
      HI.INDEX FIELD.OFFSET NEXT.PTR HI.PTR PATLENGTH])
  else (SETQ TAIL (CDR (SETQ PREV TAIL]))
)

```

(DEFINEQ

(AR.QUERY.COMPARE

```

[LAMBDA (QFORMWINDOW CLAUSE ANDINDEXES NEGFLG)
; Edited 16-Mar-88 12:30 by bvm
;; Comparison search. If ANDINDEXES is supplied, result is AND of them and this search. NEGFLG means search for those whose field is NOT
;; this value.
(LET* ((FIELD.NAME (CAR CLAUSE))
       (FIELD.SPEC (CDR (ASSOC FIELD.NAME AR.INDEX.FIELD.SPECS)))
       (FIELD.KEYLIST (LISTGET FIELD.SPEC 'ENUMERATED.FIELD.KEYLIST))
       VALUE)
  (if FIELD.KEYLIST
    then
      (AR.QUERY.COMPARE.ENUMERATED QFORMWINDOW CLAUSE ANDINDEXES NEGFLG FIELD.KEYLIST
      (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR))
      ; An enumerated field
    elseif (EQ FIELD.NAME 'Number%)
      then
      (AR.QUERY.NUMBER QFORMWINDOW CLAUSE ANDINDEXES NEGFLG)
      ; Had to check this first, since it's not a stored field name in the
      ; ordinary sense
    elseif (NOT (MEMB FIELD.NAME AR.INDEX.FIELD.LIST))
      then
      (AR.PROMPT.PRINT QFORMWINDOW T "Unknown field name: " FIELD.NAME)
      (ERROR!)
    elseif (EQ FIELD.NAME 'Date%)
      then
      (AR.QUERY.DATE QFORMWINDOW CLAUSE ANDINDEXES NEGFLG)
    elseif (STRPOS "Date" FIELD.NAME)
      then
      (AR.QUERY.GENERAL.DATE QFORMWINDOW CLAUSE ANDINDEXES NEGFLG)
      ; Some other kind of date comparison
    elseif (AND (FMEMB (CADR CLAUSE)
                      ' (= ~ =))
              (OR (NULL (SETQ VALUE (CADDR CLAUSE)))
                  (EQ (NCHARS VALUE)
                      0)))
      then
      (AR.QUERY.IS.EMPTY QFORMWINDOW FIELD.NAME ANDINDEXES (if (EQ (CADR CLAUSE)
                                                                    '=)
                                                                then NEGFLG
                                                                else (NOT NEGFLG)))
    else (AR.PROMPT.PRINT QFORMWINDOW T "Can't use numeric comparison on " FIELD.NAME)
          (ERROR!]))
)

```

(AR.QUERY.COMPARE.PARSE

```

[LAMBDA (QFORMWINDOW CLAUSE NEGFLG VALUEFN)
; Edited 17-Mar-88 12:43 by bvm
;; Parse a clause of the form (field.name > value1 [< value2]) into a list (op1 lo.num op2 hi.num negflg), where op1 is one of =, > or >=, op2 is nil or
;; one of < or <=, and negflg asks for negation. The numbers are produced by applying VALUEFN to the args value and QFORMWINDOW, and
;; must be integers. Operators may be negated in order to assure that there is always a lower bound. Complains (and aborts) if clause is
;; malformed.
(DESTRUCTURING-BIND (OP1 NUM1 . REST)
  (CDR CLAUSE)
  (LET (OP2 NUM2)
    (if (AND (FIXP (SETQ NUM1 (CL:FUNCALL VALUEFN NUM1 OP1 QFORMWINDOW)))
            (SELECTQ OP1
              (> >=)
              (< <=)
              (OR (NULL REST)

```

```

      (SELECTQ (SETQ OP2 (pop REST))
        ((< <=) ; Ok, is a between in the form > lo < hi
          (FIXP (SETQ NUM2 (CL:FUNCALL VALUEFN (pop REST)
            OP2 QFORMWINDOW))))
          NIL)))
    ((< <=) ; Reverse of above
     (if (NULL REST)
      then (SETQ OP1 (SELECTQ OP1
        (< '>=)
        '>))
        (SETQ NEGFLG (NOT NEGFLG))
        ; Reverse op and reverse negator
      else T ; Canonicalize to > lo < hi
        (SETQ OP2 OP1)
        (SETQ NUM2 NUM1)
        (SELECTQ (SETQ OP1 (pop REST))
          ((> >=)
            (FIXP (SETQ NUM1 (CL:FUNCALL VALUEFN (pop REST)
              OP1 QFORMWINDOW))))
          NIL)))
    (= T) ; Odd query--here for completeness
    (~= (SETQ NEGFLG (NOT NEGFLG)) ; Turn into = with reverse sense
      (SETQ OP1 '=))
    NIL)
  (NULL REST))
then (LIST OP1 NUM1 OP2 NUM2 NEGFLG)
else (AR.BAD.QUERY QFORMWINDOW CLAUSE])

```

(AR.QUERY.NUMBER

```

[LAMBDA (QFORMWINDOW CLAUSE ANDINDEXES NEGFLG) ; Edited 16-Mar-88 11:39 by bvm
  (DESTRUCTURING-BIND (OP.LO LO.NUM OP.HI HI.NUM NEGFLG)
    (AR.QUERY.COMPARE.PARSE QFORMWINDOW CLAUSE NEGFLG (FUNCTION CL:IDENTITY))
    (PROG (LO.INDEX HI.INDEX EXACT)
      (if (AND HI.NUM (< HI.NUM LO.NUM))
        then (AR.PROMPT.PRINT QFORMWINDOW CLAUSE " specifies a null interval."
          (ERROR!))
        (AR.PROMPT.PRINT QFORMWINDOW "Number: ")
        (CL:MULTIPLE-VALUE-SETQ (LO.INDEX EXACT)
          (AR.INDEX.FROM.NUMBER QFORMWINDOW LO.NUM))
        (SELECTQ OP.LO
          (> (if EXACT
            then ; Don't want to include LO.INDEX
              (add LO.INDEX 1))
            (= (RETURN (AR.QUERY.COMBINE.RESULT (AND EXACT (LIST LO.INDEX))
              ANDINDEXES NEGFLG)))
            NIL)
          (if OP.HI
            then (CL:MULTIPLE-VALUE-SETQ (HI.INDEX EXACT)
              (AR.INDEX.FROM.NUMBER QFORMWINDOW HI.NUM LO.INDEX))
              (if (OR (EQ OP.HI '<)
                (NOT EXACT))
                then ; Don't want to include HI.INDEX. Note that if EXACT is false,
                  ; then the index returned is that of the next highest existing AR,
                  ; or max.index+1 if out of range
                    (SETQ HI.INDEX (SUB1 HI.INDEX))
                    (RETURN (AR.QUERY.PRODUCE.INDEXES LO.INDEX HI.INDEX ANDINDEXES NEGFLG]))
              NIL)
          NIL)
    (RETURN (AR.QUERY.PRODUCE.INDEXES LO.INDEX HI.INDEX ANDINDEXES NEGFLG]))

```

(AR.QUERY.PRODUCE.INDEXES

```

[LAMBDA (LO.INDEX HI.INDEX ANDINDEXES NEGFLG) ; Edited 25-Jul-88 15:34 by bvm
  ;; Produce all indexes in range [lo,hi] (or its complement if NEGFLG is true) that are in ANDINDEXES (default everything). HI.INDEX may be NIL.
  (if (NULL ANDINDEXES)
    then ; Nothing to intersect
      (if NEGFLG
        then (NCONC (AR.COLLECT.N 0 (SUB1 LO.INDEX))
          (AND HI.INDEX (AR.COLLECT.N (ADD1 HI.INDEX)
            AR.MAX.INDEX)))
        else (AR.COLLECT.N LO.INDEX (OR HI.INDEX AR.MAX.INDEX)))
    else (LET ((TAIL ANDINDEXES)
      HI.PREV LO.PREV)
      [while (AND TAIL (< (CAR TAIL)
        LO.INDEX))
        do (SETQ TAIL (CDR (SETQ LO.PREV TAIL)) ; (CAR TAIL) is first candidate ar
          (if HI.INDEX
            then (SETQ HI.PREV LO.PREV)
              [while (AND TAIL (<= (CAR TAIL)
                HI.INDEX))
                do (SETQ TAIL (CDR (SETQ HI.PREV TAIL))
                  else (SETQ TAIL NIL))
              NIL)
          NIL)
      ;; At this point we have LO.PREV => first good ar ... HI.PREV => TAIL
      (if NEGFLG

```

```

then (if LO.PREV
      then
        (RPLACD LO.PREV TAIL)
        ANDINDEXES
      else
        TAIL)
else (if HI.PREV
      then
        (RPLACD HI.PREV NIL))
(if LO.PREV
    then
      (CDR LO.PREV)
    elseif (OR HI.PREV (NULL HI.INDEX))
    then
      ANDINDEXES
    else
      NIL))

```

; Take everything up to LO.PREV concatenated with TAIL
; Snip out middle
; Just TAIL
; Snip off tail, take everything after LO.PREV
; Found a lower bound, so take what's after it. In this case we
; have always snipped off the too-large segment
; First one satisfied lower bound and did not violate upper bound
; First one was too large

(AR.COLLECT.N

```

[LAMBDA (LO HI)
  ;; Collect the integers from LO to HI
  (for I from LO to HI collect I)]

```

; Edited 15-Mar-88 19:19 by bvm

(AR.INDEX.FROM.NUMBER

```

[LAMBDA (QFORMWINDOW NUM LO.HINT HI.HINT)
  ;; Find the index that corresponds to NUM. If we find the exact number, we return T as a second value, else NIL. LO.HINT and HI.HINT are
  ;; optional indexes known to bound the search.

```

```

(PROG ((LO.INDEX (OR LO.HINT 0))
       (HI.INDEX (OR HI.HINT AR.MAX.INDEX))
       (BOUND MID.INDEX MID.NUM))

```

;; We will do binary search over the index table.

```

[if (>= NUM (SETQ BOUND (AR.ENTRY.VALUE.FROM.INDEX HI.INDEX)))
  then
    (RETURN (if (EQ NUM BOUND)
                then
                  (CL:VALUES HI.INDEX T)
                else
                  (if HI.HINT
                    then (SHOULDNT "AR# greater than upper bound")
                    (CL:VALUES (ADD1 HI.INDEX)
                               NIL))))
  elseif (<= NUM (SETQ BOUND (AR.ENTRY.VALUE.FROM.INDEX LO.INDEX)))
  then
    (RETURN (CL:VALUES LO.INDEX (if (EQ NUM BOUND)
                                    then
                                      T
                                    else
                                      (if LO.HINT
                                        then (SHOULDNT "AR# less than upper bound")
                                        NIL)]

```

```

LP (SETQ MID.INDEX (+ LO.INDEX (IQUOTIENT (- HI.INDEX LO.INDEX)
                                           2)))

```

```

(if (EQ MID.INDEX LO.INDEX)
    then
      (RETURN (CL:VALUES HI.INDEX NIL)))
(if (< NUM (SETQ MID.NUM (AR.ENTRY.VALUE.FROM.INDEX MID.INDEX)))
    then
      (SETQ HI.INDEX MID.INDEX)
    elseif (EQ NUM MID.NUM)
    then
      (RETURN (CL:VALUES MID.INDEX T))
    else
      (SETQ LO.INDEX MID.INDEX))
(GO LP)]

```

; We made no progress, so return the next higher index
; Shot too high
; Shot too low

)

(DEFINEQ

(AR.QUERY.DATE

```

[LAMBDA (QFORMWINDOW CLAUSE ANDINDEXES NEGFLG)
  ;; Calculate range of ARs satisfying the date specification. Assume dates are monotonic

```

```

(DESTRUCTURING-BIND (LO.DATE HI.DATE NEGFLG)
  (AR.QUERY.PARSE.DATES QFORMWINDOW CLAUSE NEGFLG)
  (LET [(FIELD.SPEC (CDR (ASSOC (CAR CLAUSE)
                                AR.INDEX.FIELD.SPECS))]
        (AR.QUERY.PRODUCE.INDEXES (AR.INDEX.FROM.DATE QFORMWINDOW LO.DATE FIELD.SPEC)
                                   (AND HI.DATE (AR.INDEX.FROM.DATE QFORMWINDOW HI.DATE FIELD.SPEC T))
                                   ANDINDEXES NEGFLG])

```

)

(AR.QUERY.GENERAL.DATE

[LAMBDA (QFORMWINDOW CLAUSE ANDINDEXES NEGFLG)

; Edited 25-Jul-88 15:35 by bvm

:: Query on a date field where we can't assume dates are monotonic

```
(DESTRUCTURING-BIND
 (LO.DATE HI.DATE NEGFLG)
 (AR.QUERY.PARSE.DATES QFORMWINDOW CLAUSE NEGFLG)
 (LET*
  ((FIELD.SPEC (CDR (ASSOC (CAR CLAUSE)
                          AR.INDEX.FIELD.SPECS)))
   (OFFSET (LISTGET FIELD.SPEC 'FIELD.OFFSET))
   (BEGIN (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR))
   (TOTALSIZE (- (LISTGET FIELD.SPEC 'FIELD.END.PTR)
                 BEGIN))
   (STREAM AR.INDEX.FILE)
   LASTLENGTH STR DT)
  (if ANDINDEXES
```

then

; Only look at these ARs. Gather up the shapes all at once, so
; we can access the file efficiently.

```
(LET ((SHAPES (AR.COLLECT.SHAPES ANDINDEXES OFFSET TOTALSIZE))
      LASTPTR LEN)
  [SETFILEPTR STREAM (+ BEGIN (SETQ LASTPTR (CAAR SHAPES))
                        (for INDEX in ANDINDEXES as PAIR in SHAPES collect INDEX
                          unless (EQ (AND (if (> (SETQ LEN (CADR PAIR))
                                                0)
                                         then
                                           (\INCFILEPTR STREAM (- (CAR PAIR)
                                                                    LASTPTR))
                                           [if (NEQ LEN LASTLENGTH)
                                             then (SETQ STR (ALLOCSTRING (SETQ LASTLENGTH LEN)
                                                                           (AIN STR 1 LEN STREAM)
                                                                           (SETQ LASTPTR (+ (CAR PAIR)
                                                                    LEN))
                                                                           (SETQ DT (IDATE STR)))
                                             (> DT LO.DATE)
                                             (OR (NULL HI.DATE)
                                                  (< DT HI.DATE)))
                                             NEGFLG))]
      else (LET ((MAX.INDEX AR.MAX.INDEX)
                (for LO.INDEX from 0 to MAX.INDEX by 500
                  join (for INDEX from LO.INDEX as LEN in (PROG1 (AR.COLLECT.SIZES LO.INDEX
                                                                    (MIN (+ LO.INDEX 499)
                                                                    MAX.INDEX)
                                                                    OFFSET MAX.INDEX TOTALSIZE)
                                                                (SETFILEPTR STREAM (+ BEGIN (
```

AR.ENTRY.VALUE.FROM.INDEX
LO.INDEX OFFSET)))

collect INDEX **unless** (EQ (AND (**if** (> LEN 0)

then ; No need to advance file pointer in this loop, since all fields are
; consecutive.

```
[if (NEQ LEN LASTLENGTH)
  then (SETQ STR (ALLOCSTRING (SETQ LASTLENGTH
                                LEN)
                              (AIN STR 1 LEN STREAM)
                              (SETQ DT (IDATE STR)))
        (> DT LO.DATE)
        (OR (NULL HI.DATE)
             (< DT HI.DATE)))
  NEGFLG]]
```

(AR.QUERY.PARSE.DATES

[LAMBDA (QFORMWINDOW CLAUSE NEGFLG)

; Edited 21-Mar-88 16:35 by bvm

:: Parse a date query CLAUSE into a list (lo.date hi.date negflg), with hi.date possibly nil.

```
(DESTRUCTURING-BIND
 (OP.LO LO.DATE OP.HI HI.DATE NEGFLG)
 [AR.QUERY.COMPARE.PARSE
  QFORMWINDOW CLAUSE NEGFLG
  (FUNCTION (LAMBDA (STR OP QFORMWINDOW)
    (LET* [TIME YEAR (DT (OR (IDATE STR)
                             (PROGN
```

; try defaulting the time. Whether beginning or end of day
; depends on the comparison operator

```
(IDATE (CONCAT STR (SETQ TIME (SELECTQ OP
                                  ((> = ~)
                                   " 0:00:00")
                                  ((= >)
                                   " 23:59:59")
                                  (SHOULDNT]
  (if (OR DT (if (SETQ DT (IDATE (CONCAT STR " " (CL:MULTIPLE-VALUE-BIND (S M H D O Y)
                                                                              (CL:GET-DECODED-TIME)
                                                                              (SETQ YEAR Y))
                                                                              TIME)))
```

; Default to beginning of day

; Get end of day

then

; Succeeded by defaulting the year, too. If this is in the future,
; however, make it be last year

```

[if [AND (> (- DT (IDATE))
           (TIMES 60 60 24))
      (SETQ DT (IDATE (CONCAT STR " " (SUB1 YEAR)
                                TIME))
              (AR.PROMPT.PRINT QFORMWINDOW "[ = " (GDATE DT (DATEFORMAT
                                                         NO.SECONDS]
                                                         DT))
              then (SELECTQ OP
                        (<=
                          (ADD1 DT))
                        ((>= =)
                          (SUB1 DT))
                        DT]
                    ; Asked to include this time, so bump by a second to make an
                    ; exclusive bound

```

;; Since the code above has already arranged that the dates are exclusive bounds, we don't need to look at the operators at all, except to check
 ;; for the silly =.

```

(if (AND HI.DATE (< HI.DATE LO.DATE))
  then (AR.PROMPT.PRINT QFORMWINDOW CLAUSE " specifies a null interval.")
      (ERROR!))
(AR.PROMPT.PRINT QFORMWINDOW (CAR CLAUSE)
 " ")
(LIST LO.DATE [OR HI.DATE (if (EQ OP.LO '=)
                              then
                                ; Shorthand for anytime this day. Assume user didn't specify the
                                ; hour.
                                (+ LO.DATE (CONSTANT (ADD1 (TIMES 60 60 24]
                              NEGFLG])

```

(AR.INDEX.FROM.DATE

[LAMBDA (QFORMWINDOW DATE FIELD.SPEC UPPER.BOUNDP) ; Edited 25-Jul-88 15:36 by bvm

;; Find the index whose date value is closest to DATE--if UPPER.BOUNDP then we return the largest index whose date does not exceed DATE,
 ;; otherwise the smallest index whose date is not less than DATE. Return NIL if no such index exists.

```

(PROG ((LO.INDEX 0)
      (HI.INDEX AR.MAX.INDEX)
      (BOUND MID.INDEX MID.DATE))
  ;; We will do binary search over the index table.
  (until (SETQ BOUND (AR.DATE.FROM.INDEX HI.INDEX FIELD.SPEC)) do
    ; Just in case we can't find the dates
    (SETQ HI.INDEX (SUB1 HI.INDEX)))
  (if (> DATE (SETQ BOUND (AR.DATE.FROM.INDEX HI.INDEX FIELD.SPEC)))
    then
      ; All AR's have dates less than this, so succeed here if we
      ; wanted an upper bound
      (RETURN (AND UPPER.BOUNDP HI.INDEX)))
  (until (SETQ BOUND (AR.DATE.FROM.INDEX LO.INDEX FIELD.SPEC)) do
    ; Just in case we can't find the dates
    (SETQ LO.INDEX (ADD1 LO.INDEX)))
  (if (< DATE (SETQ BOUND (AR.DATE.FROM.INDEX LO.INDEX FIELD.SPEC)))
    then
      ; All AR's have dates greater than this, so succeed here if we
      ; wanted a lower bound
      (RETURN (AND (NOT UPPER.BOUNDP)
                  LO.INDEX)))
  LP

```

;; Invariant: desired date is always between the dates of LO.INDEX and HI.INDEX

```

(SETQ MID.INDEX (+ LO.INDEX (IQUOTIENT (- HI.INDEX LO.INDEX)
                                       2)))
(if (EQ MID.INDEX LO.INDEX)
  then
    ; At this point, LO.INDEX = HI.INDEX-1, so return one of them, depending on which side we want the date.
    (RETURN (if UPPER.BOUNDP
              then LO.INDEX
              else HI.INDEX)))
NEWDATE
(if (NULL (SETQ MID.DATE (AR.DATE.FROM.INDEX MID.INDEX FIELD.SPEC)))
  then
    ; Grumble, a dateless AR. This ought not happen
    (if (EQ (add MID.INDEX 1)
            HI.INDEX)
      then
        ; No ar's between original mid and hi have dates, so just lower hi
        ; to mid and loop
        (SETQ HI.INDEX (+ LO.INDEX (IQUOTIENT (- HI.INDEX LO.INDEX)
                                               2)))
        (GO LP)
      else (GO NEWDATE)))
(if (< DATE MID.DATE)
  then
    ; Shot too high
    (SETQ HI.INDEX MID.INDEX)
  else
    ; Shot too low
    (SETQ LO.INDEX MID.INDEX)
  (GO LP])

```

(AR.DATE.FROM.INDEX

[LAMBDA (INDEX FIELD.SPEC) ; Edited 25-Jul-88 15:36 by bvm

```

(LET* ([START (AR.ENTRY.VALUE.FROM.INDEX INDEX (LISTGET FIELD.SPEC 'FIELD.OFFSET])

```



```

(LENGTH (- (if (EQ INDEX AR.MAX.INDEX)
                then (- (LISTGET FIELD.SPEC 'FIELD.END.PTR)
                        (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR))
                else (AR.ENTRY.VALUE.NEXT))
          START))
STR DT)
(if (NEQ LENGTH 0)
    then [SETFILEPTR AR.INDEX.FILE (+ START (LISTGET FIELD.SPEC 'FIELD.BEGIN.PTR)
                                       (AIN (SETQ STR (ALLOCSTRING LENGTH))
                                             1 LENGTH AR.INDEX.FILE)
                                       (if (AND (SETQ DT (IDATE STR))
                                             (> DT 0))
                                           then
; Insist that dates be reasonable. Thus we ignore ARs submitted
; on machines whose clocks were reset to zero. Date 0 is
; actually in 1969.
DT])
    ]
)

```

(DEFINEQ

(AR.NUMS.FROM.QUERY

```

[LAMBDA (QFORMWINDOW) ; Edited 15-Jun-90 11:08 by jds
; 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.
(COND
  ((OR QFORMWINDOW (SETQ QFORMWINDOW (AR.SELECT.WINDOW "Select Query form window")))
   (WITH.AR.QUERY QFORMWINDOW (AR.ENSURE.QUERY.DATA QFORMWINDOW '(Number%:))
    (for ENTRY in (WINDOWPROP QFORMWINDOW 'AR.ENTRIES) collect (fetch (ARQUERYDATA ARQ#) of ENTRY]))
)

```

(AR.ENTRY.PTR.FROM.INDEX

```

[LAMBDA (INDEX OFFSET) ; Edited 25-Feb-88 12:40 by bvm
; Get file pointer for the OFFSET entry of AR specified by INDEX. OFFSET defaults to zero, which points at the AR number.
(if OFFSET
    then (+ (TIMES INDEX AR.INDEX.ENTRY.SIZE)
            OFFSET AR.INDEX.ENTRY.BEGIN.PTR)
    else
(+ (TIMES INDEX AR.INDEX.ENTRY.SIZE)
   AR.INDEX.ENTRY.BEGIN.PTR])
; Avoid the extra box when OFFSET is zero
)

```

(AR.ENTRY.VALUE.FROM.INDEX

```

[LAMBDA (INDEX FIELD.OFFSET) ; Edited 11-Mar-88 18:15 by bvm
; Return the 32-bit value stored at OFFSET (default zero, which is the ar #) in INDEX's fixed-size entry. Leaves file pointer positioned after having
; read that value, if you care. (Each INDEX has a table AR.INDEX.ENTRY.SIZE long of 4-byte values.)
(SETFILEPTR AR.INDEX.FILE (if FIELD.OFFSET
    then (+ (TIMES INDEX AR.INDEX.ENTRY.SIZE)
            FIELD.OFFSET AR.INDEX.ENTRY.BEGIN.PTR)
    else
; Avoid the extra box when OFFSET is zero
(+ (TIMES INDEX AR.INDEX.ENTRY.SIZE)
   AR.INDEX.ENTRY.BEGIN.PTR)))
(\DWIN AR.INDEX.FILE])
)

```

(AR.ENTRY.VALUE.NEXT

```

[LAMBDA NIL ; Edited 17-Mar-88 12:17 by bvm
; Called immediately after a call to AR.ENTRY.VALUE.FROM.INDEX, this returns the value of the next entry. Index must not have been
; max.index.
(\INCFILEPTR AR.INDEX.FILE (- AR.INDEX.ENTRY.SIZE AR.BYTES.PER.PTR))
(\DWIN AR.INDEX.FILE])
)

```

(AR.SELECT.WINDOW

```

[LAMBDA (PROMPT) ; Edited 23-Feb-88 18:56 by bvm
; Prompt user for a window with PROMPT. Returns the main window associated with window pointed to, or NIL if pointed outside a window
(PROMPTPRINT PROMPT)
(CL:UNWIND-PROTECT
 (LET [(W (WHICHW (GETPOSITION)
                  (AND W (MAINWINDOW W))))
       (CLRSPROMPT)])
)
)

```

;; Patch for nasty bug in \INCFILEPTR

(DEFINEQ

(AR.INCFILEPTR

```

[LAMBDA (STREAM AMOUNT) ; Edited 15-Jun-90 11:20 by jds
(\CALLME '\PAGED.INCFILEPTR)
; Increment file pointer of stream by AMOUNT, which may be negative. The only reason this function currently exists is to give fast performance to
; FFILEPOS -- it avoids the boxing that would occur on large file pointers.
)

```

```
(UNINTERRUPTABLY
  (PROG ((NEWOFF (+ (fetch (STREAM COFFSET) of STREAM)
                    AMOUNT))
        (NEWPAGE (fetch (STREAM CPAGE) of STREAM)))
  ;; SETFILEPTR sets CHARPOSITION to zero, but callers of \INCFILEPTR don't care, by fiat
  (COND
    ((>= NEWOFF BYTESPERPAGE) ; New page
     (SETQ NEWPAGE (+ NEWPAGE (fetch (BYTEPTR PAGE) of NEWOFF)))
     (SETQ NEWOFF (fetch (BYTEPTR OFFSET) of NEWOFF)))
    (< NEWOFF 0) ; New page going backward
     [SETQ NEWPAGE (- NEWPAGE (fetch (BYTEPTR PAGE) of (SETQ NEWOFF (SUB1 (- BYTESPERPAGE NEWOFF))
     (COND
       ((< NEWPAGE 0) ; Probably shouldn't happen; should it be an error?
        (SETQ NEWPAGE 0))]
     (SETQ NEWOFF (SUB1 (- BYTESPERPAGE (fetch (BYTEPTR OFFSET) of NEWOFF]
    ([COND
      ((< AMOUNT 0) ; Backing up, may have to set the eof if we have been writing
       (\UPDATEOF STREAM)
       T)
      T) ; Moving forward, make sure we don't move past the eof
       (AND (fetch (STREAM CBUFPTR) of STREAM)
            (<= NEWOFF (fetch (STREAM CBUFSIZE) of STREAM)
            ; easy case, no page turn
            ; Just bump COFFSET and we're done
            (replace (STREAM COFFSET) of STREAM with NEWOFF)
            (RETURN))
      T) ; Moving forward past eof, might as well let this fall thru to
         ; general case, since we need to make sure current buffer is
         ; released.
    ))
    (\UPDATEOF STREAM)
    (\RELEASECPAGE STREAM)
    (replace (STREAM CPAGE) of STREAM with NEWPAGE)
    (replace (STREAM COFFSET) of STREAM with NEWOFF))))))
)
```

```
(DECLARE%: DONTEVAL@LOAD DOCOPY
```

```
(AND (CCODEP 'AR.INCFILEPTR (MOVD 'AR.INCFILEPTR '\PAGED.INCFILEPTR NIL T))
)
```

;; Set up file names. We use VARS on AR.INDEX.DEFAULT.FILE.NAME to force it correct in the case where the index is moving. If user has set it to
 ;; some disk file for manual caching, make that the cache name

```
(RPAQ? AR.INDEX.CACHE.FILE.NAME (AND (BOUNDP 'AR.INDEX.DEFAULT.FILE.NAME)
  (STRPOS "DSK" (UNPACKFILENAME.STRING AR.INDEX.DEFAULT.FILE.NAME
  'HOST)
  NIL NIL T NIL UPPERARRAY)
  AR.INDEX.DEFAULT.FILE.NAME))
```

```
(RPAQ? AR.ALWAYS.CACHE.INDEX :ASK)
```

```
(RPAQ AR.INDEX.DEFAULT.FILE.NAME "{AR:MV:Envos}<LispARs>AR.INDEX")
```

```
(RPAQQ AR.QFORM.TITLEMENU NIL)
```

```
(RPAQQ AR.QFORM.FORMAT (|Query List:| CR |Sort List:| CR))
```

```
(RPAQQ AR.QFORM.SPECS
  ((|Query List:| FIELDTYPE STRING FN AR.QFORM.PROMPT.LIST.FN)
  (|Sort List:| FIELDTYPE STRING FN AR.QFORM.PROMPT.LIST.FN)
  (Query FIELDTYPE BUTON FN AR.QFORM.BUTTONFN FONT ARBUTTONFONT)
  (|Print File:| FIELDTYPE STRING)
  (Print FIELDTYPE BUTON FN AR.QFORM.BUTTONFN FONT ARBUTTONFONT)
  (|Update List:| FIELDTYPE STRING)
  (Update FIELDTYPE BUTON FN AR.QFORM.BUTTONFN FONT ARBUTTONFONT)
  (|Print Index Stats| FIELDTYPE BUTON FN AR.QFORM.BUTTONFN FONT ARBUTTONFONT)
  (Debug FIELDTYPE BUTON FN AR.QFORM.BUTTONFN FONT ARBUTTONFONT)))
```

```
(RPAQQ AR.QFORM.ICON
```



```
(RPAQQ AR.COMPARISON.OPERATORS (> >= < <= = ~= btwn))
```

```
(RPAQ? AR.BROWSER.MENU.ITEMS ' ("Display" AR.BROWSER.DISPLAY "Display selected AR in a readonly window")
  ("Edit" AR.BROWSER.EDIT "Edit selected AR in an AReedit window
    (uses same window as last time unless you select with middle button).")
  ("Hardcopy AR(s)" AR.BROWSER.HARDCOPY "Make hardcopy of the complete content of
    the selected AR(s)"))
```

```
(RPAQ? AR.QUERY.MENU.ITEMS ' ("Query" (AR.QFORM.QUERY)
  "Search the AR database for ARs matching the Query List")
  ("Sort" AR.QFORM.SORT "Sort the ARs in the browser window using the new Sort
  List")
  ("Hardcopy Summary" AR.QFORM.SUMMARY "Print to your default printer a summary of
  the ARs displayed in the browser" (SUBITEMS ("Text Summary"
  AR.QFORM.SUMMARY.TEXT
  "Make a plain text
  version of the summary
  on a file")
  ("TEdit Summary"
  AR.QFORM.SUMMARY.TEDIT
  "Edit (using TEdit) a plain
  text version of the
  summary"])
```

(RPAQ? AR.WHENSELECTEDSHADE 4672)

```
(RPAQ? AR.DISPLAY.FIELDS ' ((Status%: 5)
  (Subject%: 50)
  (Attn%: 15)
  (System%: 13)
  (Subsystem%: 13))
```

```
(RPAQ? AR.SUMMARY.FIELDS
  ' ((Date%: 9 T)
  (System%: 13 T)
  (Subsystem%: 14)
  (Status%: 10 T)
  (Attn%: 13)
  (Subject%: 55)
  (Priority%: 10)
  (Difficulty%: 10)
  (Impact%: 8)
  (|Problem Type:| 13))
```

(RPAQ? AR.TEDIT.FIELDS)

(RPAQ? AR.SUMMARY.MIN.LINES 2)

(RPAQ? AR.CLEANUP.SORT.ORDER ' (System%: Subsystem%: Status%: Priority%: Impact%:))

(RPAQ? AR.SORT.EQUIVALENTS ' ((Status%: (Open Open/Unreleased)))

```
(ADDTovar AR.SORT.SPEC.ITEMS ("Standard Summary Order" [FUNCTION (LAMBDA NIL AR.CLEANUP.SORT.ORDER]
  "Sort order used by AR Cleanup when producing personal summaries."))
```

```
(ADDTovar AR.QUERY.SPEC.ITEMS ("Status is UnFixed" "(OR (Status: >= Open/Unreleased) (Status: = Incomplete))"
  "AR is somehow Open, i.e., not Fixed, Declined or Obsolete")
  ("Status is Resolved" "(AND (Status: >= Obsolete) (Status: <= Fixed)" "AR has been
  taken care of--Fixed, Declined, etc.")
  ("Mandatory" "(AND (Status: >= Open/Unreleased) (Priority: = Absolutely)
  (Problem% Type: ~= Feature))" "Non-Feature AR has priority Absolutely and
  is still open somehow"))
```

(DECLARE%: EVAL@COMPILE DONTCOPY

(DECLARE%: EVAL@COMPILE

```
(TYPE RECORD AR.INDEX.DATA (AR.INDEX.FILE AR.INDEX.ENTRY.BEGIN.PTR AR.INDEX.ENTRY.END.PTR AR.INDEX.ENTRY.SIZE
  AR.INDEX.FIELD.SPECS AR.INDEX.FIELD.LIST AR.MAX.INDEX))
```

```
(RECORD ARQUERYDATA (ARQINDEX ARQCOMPLETE . ARQALLFIELDS)
  (RECORD ARQALLFIELDS (ARQ# . ARQFIELDS))
```

```
;; Data for a single AR in the query browser.
;; ARQINDEX is the index of the AR
;; ARQCOMPLETE is true if we have filled in all the fields
;; ARQ# is the first field, the ar number
;; ARQFIELDS is the rest of the fields. Each element is either a value, a (offset length) pair in scratch file, or ? to indicate incompleteness.
)
```

```
(RECORD ARINDEXDESCR (ARINAME ARIOFFKEYS ARIBEGIN . ARIEND)
```

```
;; Descriptor for a particular index field.
;; ARINAME is name of field
;; ARIOFFKEYS is offset for string field, or list of key values for enumerated field
```

;; ARIBEGIN & ARIEND are the field BEGIN and END pointers

```
)
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS AR.QFORM.ICON AR.BROWSER.MENU.ITEMS AR.QUERY.MENU.ITEMS AR.COMPARISON.OPERATORS AR.QFORM.TITLEMENU)
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(LOCALVARS . T)
)
```

```
(DEFMACRO WITH.AR.QUERY (WINDOW &BODY BODY)
[LET [(FIELDS (REVERSE (RECORDFIELDNAMES 'AR.INDEX.DATA)
;; Establish a context in which the fields of AR.INDEX.DATA from WINDOW can be referred to as variables, even specially.
;; Note: depends on AR.INDEX.DATA being a TYPERECORD and RECORDFIELDNAMES returning the fields in reverse order. This will
;; need to change if the AR.INDEX.DATA record changes
'(WITH.MONITOR (WINDOWPROP ,WINDOW 'AR.INDEX.MONITORLOCK)
(DESTRUCTURING-BIND ,FIELDS (CDR (WINDOWPROP ,WINDOW 'AR.INDEX.DATA))
(DECLARE (SPECVARS ,@FIELDS))
(IF (NOT (OPENP AR.INDEX.FILE))
THEN (AR.INDEX.FILE.REOPEN ,WINDOW))
,@BODY))])]
)
```

```
(DEFMACRO ARSPECGET (SPECS FIELDNAME PROP)
'(LISTGET (CDR (ASSOC ,FIELDNAME ,SPECS))
,PROP))
)
```

```
[CL:PROCLAIM (CONS 'CL:SPECIAL (RECORDFIELDNAMES 'AR.INDEX.DATA)
```

```
(CASE DFNFLG
(PROP ALLPROP) ; When I load this file PROP, need to get these defs eval'd, ; grumble
[LET ((DFNFLG T))
(MAPC '(WITH.AR.QUERY ARSPECGET)
(FUNCTION (LAMBDA (FN)
(CL:EVAL (GETDEF FN 'FUNCTIONS NIL ' (NOERROR))))
)
```

;; These aren't ours, but declare them to reduce the warnings from compiler & masterscope

```
(CL:PROCLAIM '(CL:SPECIAL DEFAULTFONT DEFAULTLANDPAGEREGION)
(DECLARE%: EVAL@COMPILE
(RPAQQ AR.BYTES.PER.PTR 4)
(CONSTANTS (AR.BYTES.PER.PTR 4))
)
(FILESLOAD (SOURCE)
TABLEBROWSERDECLS)
)
(DECLARE%: EVAL@COMPILE DOCOPY
(CL:PROCLAIM '(CL:SPECIAL AR.INDEX.DEFAULT.FILE.NAME AR.INDEX.CACHE.FILE.NAME AR.ALWAYS.CACHE.INDEX
AR.QFORM.SPECS AR.QFORM.FORMAT AR.WHENSELECTEDSHADE AR.DISPLAY.FIELDS AR.SUMMARY.MIN.LINES
AR.SUMMARY.FIELDS AR.TEDIT.FIELDS AR.QUERY.SPEC.ITEMS AR.SORT.SPEC.ITEMS
AR.SORT.EQUIVALENTS)
)
(PUTPROPS ARQUERY COPYRIGHT ("Venue & Xerox Corporation" 1988 1989 1990))
)
```

FUNCTION INDEX

AR#.FROM.ITEM	9	AR.INDEX.OPEN	5	AR.QUERY.AND	21
AR.AUGMENT.QUERY.FIELDS	15	AR.INDICES.FROM.FILEPTRS	27	AR.QUERY.CLOSE/SHRINK	5
AR.BAD.QUERY	21	AR.IP.FROM.SUMMARY	18	AR.QUERY.CLOSEFN	5
AR.BROWSER.COMMANDFN	9	AR.KEYVALS.FROM.KEYLIST	16	AR.QUERY.COMBINE.RESULT	23
AR.BROWSER.DISPLAY	10	AR.MAKE.COMPARISON.STRING	8	AR.QUERY.COMPARE	28
AR.BROWSER.DO.COMMAND	9	AR.MAKE.SUMMARY.FILE	16	AR.QUERY.COMPARE.ENUMERATED	24
AR.BROWSER.EDIT	11	AR.MAKE.SUMMARY.TEXT.FILE	16	AR.QUERY.COMPARE.PARSE	28
AR.BROWSER.HARDCOPY	11	AR.NUMS.FROM.QUERY	33	AR.QUERY.DATE	30
AR.BROWSER.PRINTFN	9	AR.OPEN.IP.STREAM	17	AR.QUERY.EVAL	20
AR.BROWSER.SELECTED.ARS	10	AR.PRINT.PADDED	17	AR.QUERY.EXPANDFN	5
AR.COLLECT.ENTRY.FIELDS	14	AR.PRINT.SUMMARY	18	AR.QUERY.GENERAL.DATE	31
AR.COLLECT.N	30	AR.PRINT.SUMMARY.FIELD	19	AR.QUERY.HAS	26
AR.COLLECT.SHAPES	26	AR.QFORM.BUTTONFN	6	AR.QUERY.IS	23
AR.COLLECT.SIZES	27	AR.QFORM.CREATE	2	AR.QUERY.IS.EMPTY	25
AR.DATE.FROM.INDEX	32	AR.QFORM.CREATE.ABORT	4	AR.QUERY.IS.EXACTLY	24
AR.ENSURE.QUERY.DATA	13	AR.QFORM.GDATE	4	AR.QUERY.NAND	21
AR.ENSURE.QUERY.FIELDS	13	AR.QFORM.GET.DEFAULT.INDEX	3	AR.QUERY.NUMBER	29
AR.ENSURE.QUERY.DATA.ITEM	14	AR.QFORM.GROUP.CREATE	2	AR.QUERY.OR	23
AR.ENTRY.PTR.FROM.INDEX	33	AR.QFORM.ICONFN	5	AR.QUERY.PARSE.DATES	31
AR.ENTRY.VALUE.FROM.INDEX	33	AR.QFORM.PROMPT.LIST.FN	7	AR.QUERY.PRODUCE.INDEXES	29
AR.ENTRY.VALUE.NEXT	33	AR.QFORM.QUERY	6	AR.QUERY.SHRINKFN	5
AR.GET.BUTTON.FIELD.AS.LIST	8	AR.QFORM.SORT	11	AR.QUERY.SMALLP	20
AR.GET.QLIST.PROMPT.MENU	6	AR.QFORM.SUMMARIZE.CHECK	17	AR.QUERY.SORT.CLAUSES	21
AR.GET.SLIST.PROMPT.MENU	12	AR.QFORM.SUMMARY	16	AR.QUERY.SORT.ORDER	22
AR.INCFILEPTR	33	AR.QFORM.SUMMARY.TEDIT	17	AR.QUERY.SORT.VALUE	22
AR.INDEX.FILE.CLOSE	6	AR.QFORM.SUMMARY.TEXT	16	AR.QUERY.WHENSELECTEDFN	5
AR.INDEX.FILE.REOPEN	6	AR.QFORM.TITLEMENU	8	AR.SELECT.WINDOW	33
AR.INDEX.FROM.DATE	32	AR.QLIST.MENU.COMPARISONS	7	AR.SORT.BY	12
AR.INDEX.FROM.NUMBER	30	AR.QUERY	20	AR.SPARSE.QUERY.P	27

VARIABLE INDEX

AR.ALWAYS.CACHE.INDEX	34	AR.QFORM.FORMAT	34	AR.SORT.SPEC.ITEMS	35
AR.BROWSER.MENU.ITEMS	35	AR.QFORM.ICON	34	AR.SUMMARY.FIELDS	35
AR.CLEANUP.SORT.ORDER	35	AR.QFORM.SPECS	34	AR.SUMMARY.MIN.LINES	35
AR.COMPARISON.OPERATORS	34	AR.QFORM.TITLEMENU	34	AR.TEDIT.FIELDS	35
AR.DISPLAY.FIELDS	35	AR.QUERY.MENU.ITEMS	35	AR.WHENSELECTEDSHADE	35
AR.INDEX.CACHE.FILE.NAME	34	AR.QUERY.SPEC.ITEMS	35		
AR.INDEX.DEFAULT.FILE.NAME	34	AR.SORT.EQUIVALENTS	35		

RECORD INDEX

AR.INDEX.DATA	35	ARINDEXDESCR	35	ARQUERYDATA	35
---------------	----	--------------	----	-------------	----

MACRO INDEX

ARSPECGET	36	WITH.AR.QUERY	36
-----------	----	---------------	----

CONSTANT INDEX

AR.BYTES.PER.PTR	36
------------------	----