

File created: 10-Mar-89 17:03:42 {POOH/N}<POOH>BURWELL>LISP>ARCHIVETOOL;4

changes to: (FNS FB.DoArchiveCommands)
(VARS ARCHIVETOOLCOMS)

previous date: 1-Dec-88 14:12:21 {POOH/N}<POOH>BURWELL>LISP>ARCHIVETOOL;2

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

;;
;; Copyright (c) 1985, 1986, 1988, 1989 by Xerox Corporation. All rights reserved.

(RPAQQ **ARCHIVETOOLCOMS**
[(COMS

;;; the user's interface to the archive browser

(FNS ARCHIVEBROWSER AB)

;;; command processing functions

(FNS AB.When.Selected.Fn AB.Command.Fn AB.Delete.Command AB.Expunge.Command AB.Filter.Command
AB.Recompute.Command AB.Retrieve.Command AB.Retrieve.Directory.Command
AB.Retrieve.Renamed.Command AB.Retrieve.Renamed.Aux AB.Sort.Command AB.Undelete.Command)

;;; miscellaneous functions

(FNS AB.Set.Browser.Title AB.Iconfn AB.Closefn AB.Printfn AB.Prompt.For.Input AB.Read.Directory
AB.Subitemp AB.Make.Cedar.FileName)

;;; the user that gets retrieval requests

(INITVARS (AB.archivist "Archivist"))

;;; the structure for an archive entry

(RECORDS AB.item)

;;; the icon

(BITMAPS AB.icon AB.icon.mask)
[INITVARS (AB.titled.icon (create TITLEDICON ICON _ AB.icon MASK _ AB.icon.mask TITLEREG _
(CREATEREGION 7 8 60 24)

;;; the font for the browser, which must be a fixed pitch font for now

(VARS (AB.browser.font (FONTCREATE 'TERMINAL 10)))

;;; based on the TableBrowser package

;;; took out WORDFNS

(FILES TABLEBROWSER))
(COMS (* LOAD the FILEBROWSER first)
(FILES FILEBROWSER))
[COMS (* ArchiveTool File Browser Interface Functions)
(FNS Arch.ConvertToCedarFileName FB.ArchiveCommand FB.ArchiveAllCommand FB.ArchiveAndDeleteCommand
FB.ArchiveAndDeleteAllCommand FB.DoArchiveCommands)
(GLOBALVARS Arch.CcToSelfFlg Arch.VerifyNotArchivedFlg OKCedarCharBitTable)
[VARS (OKCedarCharBitTable (MAKEBITTABLE (APPEND (CHARCODE (%. \$ - + * < > { })))
(for i from (CHARCODE a)
to
(CHARCODE z)
collect i)
(for i from (CHARCODE A)
to
(CHARCODE Z)
collect i)
(CHARCODE (0 1 2 3 4 5 6 7 8 9]

(INITVARS (Arch.CcToSelfFlg T)
(Arch.VerifyNotArchivedFlg NIL))
(DECLARE%: DONTVAL@LOAD DOCOPY (ADDVARS (BackgroundMenuCommands (ArchiveBrowser ' (ARCHIVEBROWSER)
"Brings up an archive

```

browser")))
(P (SETQ BackgroundMenu))
(DECLARE%: EVAL@COMPILE DONTCOPY (FILES (SOURCE)
TABLEBROWSERDECLS)
(FILEBROWSER)
(FILES (LOADCOMP)
FILEBROWSER))
(DECLARE%: DONTVAL@LOAD DONTCOPY (P (if (NOT (SASSOC "Archive" FB.MENU.ITEMS))
then
(PUTASSOC "Archive" '(FB.ArchiveCommand "Archives selected
files by sending mail to
Archivist.pa"
(SUBITEMS ("Archive"
FB.ArchiveCommand
"Archives
selected files
by sending
mail to
Archivist.pa"))
("Archive ALL Files"
FB.ArchiveAllCommand
"Archives ALL files in
the browser by sending
mail to Archivist.pa")
("Archive and Delete"
FB.ArchiveAndDeleteCommand
"Archives selected
files by sending an
Archive and Delete
request to
Archivist.pa")
("Archive and Delete ALL
Files"
FB.ArchiveAndDeleteAllCommand
"Archives ALL
files in the
browser by
sending an
Archive and
Delete request to
Archivist.pa"))))
else
(RPLACD (SASSOC "Archive" FB.MENU.ITEMS)
'(FB.ArchiveCommand "Archives selected files by
sending mail to Archivist.pa"
(SUBITEMS ("Archive" FB.ArchiveCommand
"Archives selected files by
sending mail to
Archivist.pa")
("Archive ALL Files"
FB.ArchiveAllCommand
"Archives ALL files in the
browser by sending mail to
Archivist.pa")
("Archive and Delete"
FB.ArchiveAndDeleteCommand
"Archives selected files by
sending an Archive and Delete
request to Archivist.pa")
("Archive and Delete ALL Files"
FB.ArchiveAndDeleteAllCommand
"Archives ALL files in the
browser by sending an Archive
and Delete request to
Archivist.pa"]
(COMS (* Functions that aren't used any more)
(FNS Arch.ParseReturnMsg Arch.BackgroundMenuFn))
(DECLARE%: DONTVAL@LOAD DONTVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA AB)
(NLAML)
(LAMA]))

```

::: the user's interface to the archive browser

(DEFINEQ

(ARCHIVEBROWSER

```

[LAMBDA (archivefilespec filter)
(LET* [(menu (create MENU
```

(* N.H.Briggs "2-Mar-87 16:56")

```

ITEMS _ ' ((Retrieve AB.Retrieve.Command "Retrieve selected files"
(SUBITEMS ("Retrieve selected files" AB.Retrieve.Command "Retrieve
selected files")
("Retrieve to directory" AB.Retrieve.Directory.Command
"Retrieve selected files to a different directory")
("Retrieve renamed" AB.Retrieve.Renamed.Command "Retrieve
selected files specifying new name for each file"))))

```

```

(Filter AB.Filter.Command "Set filter for displayed file names")
(Sort AB.Sort.Command "Sort entries by file name" (SUBITEMS
  ("Sort by file name"
   AB.Sort.Command
   "Sort entries by
   file name")
  ("Sort by creation date"
   (AB.Sort.Command
    CreationDate)
   "Sort entries by creation
   date of the file")
  ("Sort by archive date"
   (AB.Sort.Command Archive)
   "Sort entries by date that
   the file was archived")
  (Reverse (AB.Sort.Command
            Reverse)
            "Reverse the order
            of the entries"))))
(Recompute AB.Recompute.Command "Redisplay browser items after re-reading
  archive directory" (SUBITEMS ("Same directory" AB.Recompute.Command
  "Redisplay browser items after
  re-reading archive directory")
  ("New directory" (AB.Recompute.Command T)
  "Browse a different archive
  directory"))))
("" NIL "do nothing - a separator")
(Delete AB.Delete.Command "Delete selected items")
(Undelete AB.Undelete.Command "Undelete selected items"
  (SUBITEMS ("Undelete selected items" AB.Undelete.Command "Undelete
  selected items")
  ("Undelete ALL items" (AB.Undelete.Command T)
  "Undelete all deleted items"))))
("" NIL "do nothing - a separator so you don't accidentally Expunge")
(Expunge AB.Expunge.Command "Expunge deleted items and rewrite the archive
  directory"))

CENTERFLG _ T
TITLE _ " Commands "
WHENSELECTEDFN _ 'AB.When.Selected.Fn)
(promptfont (FONTCREATE 'HELVETICA 10))
(promptheight (HEIGHTIFWINDOW (TIMES 2 (FONTPROP promptfont 'HEIGHT))
  T))

(promptwindow)
(windowregion (GETREGION (PLUS (fetch IMAGEWIDTH of menu)
  144)
  (PLUS (fetch IMAGEHEIGHT of menu)
  promptheight)))
(window (CREATEW (CREATEREGION (fetch LEFT of windowregion)
  (fetch BOTTOM of windowregion)
  (DIFFERENCE (fetch WIDTH of windowregion)
  (fetch IMAGEWIDTH of menu))
  (DIFFERENCE (fetch HEIGHT of windowregion)
  promptheight))
  ""))
(browser (TB.MAKE.BROWSER NIL window '(PRINTFN AB.Printfn FONT ,AB.browser.font]
(ATTACHMENU menu window 'RIGHT 'TOP)
(TB.USERDATA browser (LIST 'ARCHIVE (PACKFILENAME.STRING 'HOST (OR (FILENAMEFIELD archivefilespec
  'HOST)
  (FILENAMEFIELD (DIRECTORYNAME)
  'HOST))
  'DIRECTORY
  (OR (FILENAMEFIELD archivefilespec 'DIRECTORY)
  (CAR (FULLUSERNAME T)))
  'NAME
  (OR (FILENAMEFIELD archivefilespec 'NAME)
  'Archive)
  'EXTENSION
  (OR (FILENAMEFIELD archivefilespec 'EXTENSION)
  'directory)
  'BODY archivefilespec)
  'FILTER
  (OR filter "*.*)")) (* (use something like this if the "attic" is used) L-CASE
  (OR filter (CONCAT (CAR (FULLUSERNAME T)) ">*.*)")))
(SETQ promptwindow (GETPROMPTWINDOW window 2 (FONTCREATE 'HELVETICA 10)))
(AB.Set.Browser.Title browser)
(WINDOWPROP promptwindow 'MINSIZE (CONS 0 (fetch (REGION HEIGHT) of (WINDOWPROP promptwindow
  'REGION])
(WINDOWPROP promptwindow 'MAXSIZE (CONS 64000 (fetch (REGION HEIGHT) of (WINDOWPROP promptwindow
  'REGION])
(LINELENGTH MAX.SMALLP promptwindow)
(WINDOWPROP window 'ICONFN (FUNCTION AB.Iconfn))
(WINDOWADDPROP window 'CLOSEFN (FUNCTION AB.Closefn)
  T)
(AB.Command.Fn (SASSOC 'Recompute (fetch (MENU ITEMS) of menu)
  menu
  'LEFT]))

```

```
(AB
  [NLAMBDA filespec% filter
    (LET ((patternandfilter (NLAMBDA.ARGs filespec% filter)))
      (ARCHIVEBROWSER (CAR patternandfilter)
        (CADR patternandfilter)
        NIL))
  )
  (* N.H.Briggs "4-Mar-87 12:11")
```

;;; command processing functions

(DEFINEQ

```
(AB.When.Selected.Fn
  [LAMBDA (Item Menu Key)
    (if (AND (LISTP Item)
      (CADR Item))
      then (TB.PROCESS (LIST (FUNCTION AB.Command.Fn)
        (KWOTE Item)
        (KWOTE Menu)
        (KWOTE Key))
        (PACK* 'AB- (CAR Item))
      )
    )
  (* N.H.Briggs "25-Jun-86 11:48")
```

```
(AB.Command.Fn
  [LAMBDA (item menu key)
    (RESETLST
      (LET* [(realitem item)
        (window (WINDOWPROP (WFROMMENU menu)
          'MAINWINDOW))
        (browser (WINDOWPROP window 'TABLEBROWSER)
          [if (NOT (MEMBER item (fetch (MENU ITEMS) of menu)))
            then
              (SETQ item (for I in (fetch (MENU ITEMS) of menu) thereis (AB.Subitem temp item I)
                (if (OBTAIN.MONITORLOCK (fetch (TABLEBROWSER TBLOCK) of browser)
                  T T)
                  then (RESETSAVE (SHADEITEM item menu MENUSELECTSHADE)
                    (LIST (FUNCTION SHADEITEM)
                      item menu WHITESHADE))
                    (LET ((function (CADR realitem))
                      (promptwindow (GETPROMPTWINDOW window)
                        extra)
                      (if (OPENWP promptwindow)
                        then (CLEARW promptwindow))
                      (if (LISTP function)
                        then (SETQ extra (CADR function))
                          (SETQ function (CAR function)))
                      (APPLY* function browser extra))
                    else (TB.BROWSER.BUSY browser)))))]
      )
    )
  (* N.H.Briggs "18-Jun-86 13:09")
```

```
(AB.Delete.Command
  [LAMBDA (browser)
    (LET ((count 0))
      [TB.MAP.SELECTED.ITEMS browser (FUNCTION (LAMBDA (browser item)
        (TB.DELETE.ITEM browser item)
        (add count 1)
      )
      (printout (GETPROMPTWINDOW (TB.WINDOW browser))
        count " item" (if (IGREATERP count 1)
          then "s"
          else "")
        " marked for deletion.")]
    )
  (* N.H.Briggs "16-Apr-86 20:56")
```

```
(AB.Expunge.Command
  [LAMBDA (browser)
    (if (EQ (fetch (TABLEBROWSER TB#DELETED) of browser)
      0)
      then (printout (GETPROMPTWINDOW (TB.WINDOW browser))
        "Nothing to expunge!")
      else (LET [(directorystream (OPENSTREAM (PACKFILENAME 'VERSION NIL 'BODY (LISTGET (TB.USERDATA browser)
        'ARCHIVE)))
        'OUTPUT]
        (if (NOT directorystream)
          then (printout (GETPROMPTWINDOW (TB.WINDOW browser))
            "Error opening (new version of) archive directory "
            (LISTGET (TB.USERDATA browser)
              'ARCHIVE))
            "...aborted.")
          else (LINELENGTH MAX.SMALLP directorystream)
            (* ensure nothing wraps around)
            (LISTPUT (TB.USERDATA browser)
              'ALLITEMS
              (for item in (LISTGET (TB.USERDATA browser)
                'ALLITEMS)
                when (NOT (TB.ITEM.DELETED? browser item))
              )
            )
          )
        )
    )
  (* N.H.Briggs "18-Jun-86 12:23")
```

```

collect (printout directorystream (fetch (AB.item AB.FileName)
                                         of (fetch (TABLEITEM TIDATA) of item))
        %,
        (fetch (AB.item AB.Creation.Date) of (fetch (TABLEITEM TIDATA)
                                                    of item)))
(for media on (fetch (AB.item AB.Media) of (fetch (TABLEITEM TIDATA)
                                                  of item))
 by (CDDR media) do (printout directorystream %, (CAR media))
                    (printout directorystream %, (CADR media))
 finally (printout directorystream T))
item))
(LISTPUT (TB.USERDATA browser)
 'ARCHIVE
 (L-CASE (FULLNAME directorystream)))
(CLOSEF directorystream)
(AB.Set.Browser.Title browser)
(TB.MAP.DELETED.ITEMS browser (FUNCTION TB.REMOVE.ITEM])

```

(AB.Filter.Command

```

[LAMBDA (browser) (* N.H.Briggs "18-Jun-86 12:28")
 (LET ((pattern (AB.Prompt.For.Input "Files matching what? " (LISTGET (TB.USERDATA browser)
 'FILTER)
 browser T)))
 (if pattern
 then (LISTPUT (TB.USERDATA browser)
 'FILTER
 (L-CASE pattern))
 (AB.Set.Browser.Title browser)
 (AB.Recompute.Command browser])

```

(AB.Recompute.Command

```

[LAMBDA (browser newdirectory?) (* N.H.Briggs "19-Sep-86 12:34")
 (LET* ((window (TB.WINDOW browser))
 (windowregion (WINDOWPROP window 'REGION))
 (region (CREATEREGION 0 0 (fetch (REGION WIDTH) of windowregion)
 (fetch (REGION HEIGHT) of windowregion)))
 (namewidth 0)
 (userdata (TB.USERDATA browser))
 [filter (DIRECTORY.MATCH.SETUP (PACKFILENAME 'BODY (LISTGET userdata 'FILTER]
 result)
 (if (NOT (ZEROP (fetch (TABLEBROWSER TB#DELETED) of browser)))
 then (if (MENU (create MENU
 ITEMS _ '(("Expunge" 'Expunge "Expunge items marked for deletion")
 ("Don't Expunge" NIL "Don't expunge items marked for deletion"))
 TITLE _ "Expunge deleted items?"
 CENTERFLG _ T)
 then (AB.Expunge.Command browser)))
 (if (AND newdirectory? (SETQ result (AB.Prompt.For.Input "New archive directory? " NIL browser T)))
 then (LISTPUT userdata 'ARCHIVE (PACKFILENAME.STRING 'NAME (OR (FILENAMEFIELD result 'NAME)
 'Archive)
 'EXTENSION
 (OR (FILENAMEFIELD result 'EXTENSION)
 'directory)
 'BODY result)))
 (if (OR (NOT newdirectory?)
 (AND newdirectory? result))
 then (TB.REPLACE.ITEMS browser)
 (LISTPUT userdata 'ALLITEMS (for item in (AB.Read.Directory browser) bind tableitem
 eachtime (SETQ tableitem (create TABLEITEM
 TIDATA _ item))
 collect (if (DIRECTORY.MATCH filter (PACKFILENAME
 'BODY
 (fetch (AB.item AB.FileName)
 of item)))
 then (SETQ namewidth
 (MAX namewidth (STRINGWIDTH (fetch (AB.item
 AB.FileName)
 of item)
 AB.browser.font)))
 (TB.INSERT.ITEM browser tableitem))
 tableitem))
 (LISTPUT userdata 'NAMEWIDTH namewidth)
 (TB.DISPLAY.LINES browser (TB.FIRST.VISIBLE.ITEM# browser region)
 (TB.LAST.VISIBLE.ITEM# browser region])

```

(AB.Retrieve.Command

```

[LAMBDA (browser) (* N.H.Briggs " 3-Mar-87 12:41")
 (DECLARE (GLOBALVARS AB.archivist))
 (PROG ((count 0)
 registry corestream)
 [SETQ registry (SELECTQ (OR (LAFITEMODE)
 (\LAFITE.INFER.MODE))
 (GV ".pa")
 (NS ":PA")

```

```

(RETURN (printout (GETPROMPTWINDOW (TB.WINDOW browser))
"Can't retrieve -- Lafite mode must be set to GV or NS"]
(SETQ corestream (OPENSTREAM '{NODIRCORE} 'BOTH))
(LINELENGTH MAX.SMALLP corestream)
(printout corestream "Subject: Retrieve request" T "To: " AB.archivist registry T "cc: " (FULLUSERNAME
)
T T)
[TB.MAP.SELECTED.ITEMS browser (FUNCTION (LAMBDA (browser item)
(printout corestream "Retrieve: "
(fetch (AB.item AB.Filename)
of (fetch (TABLEITEM TIDATA) of item))
" of "
(fetch (AB.item AB.Creation.Date)
of (fetch (TABLEITEM TIDATA) of item))
" from "
(CAR (fetch (AB.item AB.Media)
of (fetch (TABLEITEM TIDATA) of item)))
" or "
(CADR (fetch (AB.item AB.Media)
of (fetch (TABLEITEM TIDATA) of item)))
T)
(add count 1]
(SETQ corestream (OPENTEXTSTREAM corestream))
(printout (GETPROMPTWINDOW (TB.WINDOW browser))
"Sending mail to " AB.archivist registry " requesting " count " file"
(if (NEQ count 1)
then "s: "
else ": ")
(if (CAR (ERSETQ (LAFITE.SENDMESSAGE corestream)))
then (printout (GETPROMPTWINDOW (TB.WINDOW browser))
" done.")
else (printout (GETPROMPTWINDOW (TB.WINDOW browser))
" failed."))

```

(AB.Retrieve.Directory.Command

(* N.H.Briggs " 3-Mar-87 12:41")

```

[LAMBDA (browser)
(DECLARE (GLOBALVARS AB.archivist))
(PROG ((count 0)
registry corestream newdirectory)
[SETQ registry (SELECTQ (OR (LAFITEMODE)
(\LAFITE.INFER.MODE))
(GV ".pa")
(NS ":PA")
(RETURN (printout (GETPROMPTWINDOW (TB.WINDOW browser))
"Can't retrieve -- Lafite mode must be set to GV or NS"]
(if (NOT (SETQ newdirectory (AB.Prompt.For.Input "Directory to retrieve into? " NIL browser T)))
then (RETURN))
(SETQ corestream (OPENSTREAM '{NODIRCORE} 'BOTH))
(LINELENGTH MAX.SMALLP corestream)
(printout corestream "Subject: Retrieve request" T "To: " AB.archivist registry T "cc: " (FULLUSERNAME
)
T T)
[TB.MAP.SELECTED.ITEMS browser (FUNCTION (LAMBDA (browser item)
(printout corestream "Retrieve: "
(fetch (AB.item AB.Filename)
of (fetch (TABLEITEM TIDATA) of item))
" of "
(fetch (AB.item AB.Creation.Date)
of (fetch (TABLEITEM TIDATA) of item))
" from "
(CAR (fetch (AB.item AB.Media)
of (fetch (TABLEITEM TIDATA) of item)))
" or "
(CADR (fetch (AB.item AB.Media)
of (fetch (TABLEITEM TIDATA) of item)))
" as "
[AB.Make.Cedar.Filename
(PACKFILENAME.STRING 'DIRECTORY newdirectory
'VERSION NIL 'BODY
(fetch (AB.item AB.Filename)
of (fetch (TABLEITEM TIDATA) of item))
T)
(add count 1]
(SETQ corestream (OPENTEXTSTREAM corestream))
(printout (GETPROMPTWINDOW (TB.WINDOW browser))
"Sending mail to " AB.archivist registry " requesting " count " file"
(if (NEQ count 1)
then "s: "
else ": ")
(if (CAR (ERSETQ (LAFITE.SENDMESSAGE corestream)))
then (printout (GETPROMPTWINDOW (TB.WINDOW browser))
" done.")
else (printout (GETPROMPTWINDOW (TB.WINDOW browser))
" failed."))

```

(AB.Retrieve.Renamed.Command

(* N.H.Briggs " 3-Mar-87 12:34")

```
[LAMBDA (browser)
  (DECLARE (GLOBALVARS AB.archivist))
  (PROG ((count 0)
    registry corestream)
    (DECLARE (SPECVARS corestream count))
    [SETQ registry (SELECTQ (OR (LAFITEMODE)
      (\LAFITE.INFER.MODE))
      (GV ".pa")
      (NS ":PA")
      (RETURN (printout (GETPROMPTWINDOW (TB.WINDOW browser))
        "Can't retrieve -- Lafite mode must be set to GV or NS"]
      (SETQ corestream (OPENSTREAM '{NODIRCORE}' BOTH))
      (LINELENGTH MAX.SMALLP corestream)
      (printout corestream "Subject: Retrieve request" T "To: " AB.archivist registry T "cc: " (FULLUSERNAME
        )
        T T)
      (TB.MAP.SELECTED.ITEMS browser (FUNCTION AB.Retrieve.Renamed.Aux))
      (if (EQ count 0)
        then (printout (GETPROMPTWINDOW (TB.WINDOW browser))
          "Nothing to retrieve."
          (CLOSEF corestream)
          (RETURN))
        (SETQ corestream (OPENTEXTSTREAM corestream))
        (printout (GETPROMPTWINDOW (TB.WINDOW browser))
          "Sending mail to " AB.archivist registry " requesting " count " file"
          (if (NEQ count 1)
            then "s: "
            else ": "))
        (if (CAR (ERSETQ (LAFITE.SENDMESSAGE corestream)))
          then (printout (GETPROMPTWINDOW (TB.WINDOW browser))
            " done.")
          else (printout (GETPROMPTWINDOW (TB.WINDOW browser))
            " failed."))])])
```

(AB.Retrieve.Renamed.Aux

(* N.H.Briggs " 3-Mar-87 12:38")

```
[LAMBDA (browser item)
  (LET ((newname (AB.Prompt.For.Input (CONCAT "Retrieve " (fetch (AB.item AB.Filename) of (fetch (TABLEITEM TIDATA)
    of item))
    " as ?")
    (fetch (AB.item AB.Filename) of (fetch (TABLEITEM TIDATA) of item))
    browser "... skipped")))
    (if newname
      then (printout corestream "Retrieve: " (fetch (AB.item AB.Filename) of (fetch (TABLEITEM TIDATA)
        of item))
        " of "
        (fetch (AB.item AB.Creation.Date) of (fetch (TABLEITEM TIDATA) of item))
        " from "
        (CAR (fetch (AB.item AB.Media) of (fetch (TABLEITEM TIDATA) of item)))
        " or "
        (CADR (fetch (AB.item AB.Media) of (fetch (TABLEITEM TIDATA) of item)))
        " as "
        (AB.Make.Cedar.Filename (PACKFILENAME.STRING 'VERSION NIL 'BODY newname))
        T)
      (add count 1))
    (HELP]))
```

(AB.Sort.Command

(* N.H.Briggs "17-Jun-86 12:47")

```
[LAMBDA (browser sorttype)
  (LET ((items (fetch (TABLEBROWSER TBITEMS) of browser))
    [if (EQ sorttype 'Reverse)
      then (SETQ items (DREVERSE items))
      else (SORT items (SELECTQ sorttype
        (CreationDate [FUNCTION (LAMBDA (x y)
          (IGREATERP (IDATE (fetch (AB.item AB.Creation.Date)
            of (fetch (TABLEITEM TIDATA)
              of x)))
          (IDATE (fetch (AB.item AB.Creation.Date)
            of (fetch (TABLEITEM TIDATA)
              of y]))
        (Archive [FUNCTION (LAMBDA (x y)
          (ILESSP (fetch (AB.item AB.Sequence.Number)
            of (fetch (TABLEITEM TIDATA) of x))
          (fetch (AB.item AB.Sequence.Number)
            of (fetch (TABLEITEM TIDATA) of y]))
        (FUNCTION (LAMBDA (x y)
          (ALPHORDER (fetch (AB.item AB.Filename) of (fetch (TABLEITEM TIDATA)
            of x))
          (fetch (AB.item AB.Filename) of (fetch (TABLEITEM TIDATA)
            of y)))
          (UPPERCASEARRAY]
    (for item in items as i from 1 do (replace TI# of item with i))
    (TB.REPLACE.ITEMS browser items)
    (TB.REDISPLAY.ITEMS browser])
```

(AB.Undelete.Command

```
[LAMBDA (browser all?) (* N.H.Briggs "16-Apr-86 21:01")
  (LET ((count 0)
    [if all?
      then [TB.MAP.DELETED.ITEMS browser (FUNCTION (LAMBDA (browser item)
        (TB.UNDELETE.ITEM browser item)
          (add count 1]
      else (TB.MAP.SELECTED.ITEMS browser (FUNCTION (LAMBDA (browser item)
        (TB.UNDELETE.ITEM browser item)
          (add count 1]
    (printout (GETPROMPTWINDOW (TB.WINDOW browser))
      count " item" (if (NEQ count 1)
        then "s"
        else "")
      " undeleted.")]
  )
```

;;; miscellaneous functions

(DEFINEQ

(AB.Set.Browser.Title

```
[LAMBDA (browser) (* N.H.Briggs "17-Jun-86 15:45")
  (LET [(archive (LISTGET (TB.USERDATA browser)
    'ARCHIVE))
    (filter (LISTGET (TB.USERDATA browser)
    'FILTER)
    (WINDOWPROP (GETPROMPTWINDOW (TB.WINDOW browser))
    'TITLE
    (CONCAT "Archive Browser" (OR (AND archive (CONCAT " " archive))
    ""))
    (OR (AND filter (CONCAT " - files " filter))
    "")])
```

(AB.Iconfn

```
[LAMBDA (window icon) (* N.H.Briggs "19-Sep-86 18:58")
  (DECLARE (GLOBALVARS AB.titled.icon))
  (LET* [(browser (WINDOWPROP window 'TABLEBROWSER))
    (archive (LISTGET (TB.USERDATA browser)
    'ARCHIVE))
    (directory (UNPACKFILENAME.STRING archive 'DIRECTORY))
    (host (UNPACKFILENAME.STRING archive 'HOST))
    (title (PACKFILENAME.STRING 'HOST host 'DIRECTORY (SUBSTRING directory 1 (STRPOS directory ">"))
    (if icon
      then (ICONW.TITLE icon title)
      else (TITLEDICONW AB.titled.icon title (FONTCREATE 'MODERN 8)
        NIL NIL NIL (CHARCODE {}))
```

(AB.Closefn

```
[LAMBDA (window) (* edited%: "20-Jun-86 12:42")
  (LET [(browser (WINDOWPROP window 'TABLEBROWSER)
    (if (NOT (ZEROP (fetch (TABLEBROWSER TB#DELETED) of browser)))
      then (if (MENU (create MENU
        ITEMS _ ' ("Expunge" 'Expunge "Expunge items marked for deletion")
        ("Don't Expunge" NIL "Don't expunge items marked for deletion"))
        TITLE _ "Expunge deleted items?"
        CENTERFLG _ T)
      then (AB.Expunge.Command browser)))
    NIL])
```

(AB.Printfn

```
[LAMBDA (browser item window) (* N.H.Briggs "22-Sep-86 13:09")
  (LET* ((entry (fetch TIDATA of item))
    (namewidth (OR (LISTGET (TB.USERDATA browser)
    'NAMEWIDTH)
    0))
    (offset (DSPXPOSITION NIL window))
    (datestart (IPLUS offset namewidth 10)))
    (* if this is to work for variable spaced fonts it has to be smart about the widths of date and media fields too)
    (PRIN1 (fetch (AB.item AB.Filename) of entry)
      window)
    (if (ZEROP namewidth)
      then (PRIN1 " " window)
      else (BLTSHADE WHITESHAE window (DSPXPOSITION NIL window)
        (IDIFFERENCE (DSPYPOSITION NIL window)
        (FONTPROP AB.browser.font 'DESCENT))
        (IDIFFERENCE datestart (DSPXPOSITION NIL window))
        (FONTPROP AB.browser.font 'HEIGHT))
```



```

(DSPXPOSITION datestart window))
(PRIN1 (fetch (AB.item AB.Creation.Date) of entry)
window)
(PRIN1 " " window)
(for media in (fetch (AB.item AB.Media) of entry) do (PRIN1 media window)
(PRIN1 " " window])

```

(AB.Prompt.For.Input

[LAMBDA (prompt default browser abortflag) (* N.H.Briggs "22-Apr-86 17:32")

(* * Prompt for input for browser browser with question prompt offering default answer DEFAULT. If abortflag is true and response is NIL, prints "... aborted" or abortflag (should be a text string))

```

(LET* ((promptwindow (GETPROMPTWINDOW (TB.WINDOW browser)))
(promptwidth (STRINGWIDTH prompt promptwindow))
(windowwidth (WINDOWPROP promptwindow 'WIDTH))
result)
(CLEARW promptwindow)
[if (IGREATERP (IPLUS promptwidth (STRINGWIDTH (OR default "XXX")
promptwindow))
windowwidth)
then
(* Prompt plus default response will overflow the width of the window, so be a nice guy and break it up)
(for i from (IDIFFERENCE (NCHARS prompt)
4)
to 10 by -1 bind (excesswidth _ (IDIFFERENCE promptwidth windowwidth))
when (AND (EQ (NTHCHARCODE prompt i)
(CHARCODE SPACE))
(IGREATERP (STRINGWIDTH (SUBSTRING prompt i)
promptwindow)
excesswidth))
do (RETURN (SETQ prompt (CONCAT (SUBSTRING prompt 1 (IPLUS i -1))
"
"
(SUBSTRING prompt (IPLUS i 1)
[SETQ result (CAR (NLSETQ (PROMPTFORWARD prompt default NIL promptwindow NIL 'TTY (CHARCODE
(CR ESC)
(if (AND (EQ result NIL)
abortflag)
then (PRINTOUT promptwindow (if (EQ abortflag T)
then "... aborted"
else abortflag)))
(TERPRI promptwindow)
result])

```

(AB.Read.Directory

[LAMBDA (browser) (* N.H.Briggs "22-Sep-86 12:53")

```

(LET ([directorystream (AND (LISTGET (TB.USERDATA browser)
'ARCHIVE)
(CAR (NLSETQ (OPENSTREAM (LISTGET (TB.USERDATA browser)
'ARCHIVE)
'INPUT]
(linerhtable (COPYREADTABLE 'ORIG))
(promptwindow (GETPROMPTWINDOW (TB.WINDOW browser)))
items)
(if (NOT directorystream)
then (printout promptwindow "Can't find archive directory " (LISTGET (TB.USERDATA browser)
'ARCHIVE))
(LISTPUT (TB.USERDATA browser)
'ARCHIVE NIL)
(AB.Set.Browser.Title browser)
NIL
else (LISTPUT (TB.USERDATA browser)
'ARCHIVE
(L-CASE (FULLNAME directorystream)))
(AB.Set.Browser.Title browser)
(SETSEPR (LIST (CONSTANT (CHARCODE EOL)))
NIL linerhtable)
(SETBRK NIL NIL linerhtable)
(bind start end inputline repeatuntil (EOFP directorystream) eachtime (SETQ inputline
(RSTRING directorystream
linerhtable))
(READC directorystream)
as i from 1 collect (create AB.item
AB.Filename _ [L-CASE (SUBSTRING inputline 1
(SUB1 (SETQ end (STRPOS " " inputline)
AB.Creation.Date _ [SUBSTRING inputline (ADD1 end)
(SUB1 (SETQ end (STRPOS " " inputline
(ADD1 end)
AB.Media _ [first (SETQ end (ADD1 end)) repeatwhile end
eachtime (SETQ start (ADD1 end))
(while (EQUAL (SUBSTRING inputline start start)

```

```

" ")
do (SETQ start (ADD1 start))
  (SETQ end (STRPOS " " inputline start))
collect (SUBSTRING inputline start (AND end (SUB1 end)
AB.Sequence.Number _ i)
finally (CLOSEF directorystream])

```

(AB.Subitemp

```

[LAMBDA (subitem item) (* N.H.Briggs "16-Apr-86 18:32")

```

(* True if subitem appears among the subitems of item or descendents)

```

(LET ((sub (CADDR item))
      (AND sub (EQ (CAR (LISTP sub))
                    'SUBITEMS)
      (OR (MEMBER subitem sub)
          (for i in (CDR sub) thereis (AB.Subitemp subitem i)

```

(AB.Make.Cedar.FileName

```

[LAMBDA (filename) (* N.H.Briggs " 3-Mar-87 12:08")

```

```

(LET ((unpackedfilename (UNPACK filename))
      (PACK (SUBLIS '({ . %[
                    () . %]
                    (; . !))
            unpackedfilename))

```

)

;;; the user that gets retrieval requests

```

(RPAQ? AB.archivist "Archivist")

```

;;; the structure for an archive entry

```

(DECLARE%: EVAL@COMPILE
(RECORD AB.item (AB.FileName AB.Creation.Date AB.Media AB.Sequence.Number))
)

```

;;; the icon

```

(RPAQ? AB.icon

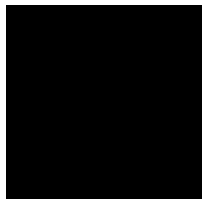
```



```

(RPAQ? AB.icon.mask

```



```

(RPAQ? AB.titled.icon (create TITLEDICON ICON _ AB.icon MASK _ AB.icon.mask TITLEREG _ (CREATEREGION 7 8 60 24)))

```

;;; the font for the browser, which must be a fixed pitch font for now

```

(RPAQ? AB.browser.font (FONTCREATE 'TERMINAL 10))

```

;;; based on the TableBrowser package

;;; took out WORDFNS

```

(FILESLoad TABLEBROWSER)

```

(* LOAD the FILEBROWSER first)

```

(FILESLoad FILEBROWSER)

```

(* ArchiveTool File Browser Interface Functions)

(DEFINEQ

(Arch.ConvertToCedarFileName

[LAMBDA (FileEntry LispFileName BROWSER)

; Edited 1-Dec-88 12:47 by bbb

;;; Convert a Lisp file name to a Cedar filename with the [host] naming convention and the ! version numbering

```

(DECLARE (GLOBALVARS OKCedarCharBitTable))
(DECLARE (SPECVARS SkippedBadFileName))
(LET* ((UnpackedName (UNPACKFILENAME.STRING LispFileName))
      (Host (LISTGET UnpackedName 'HOST))
      (Name (LISTGET UnpackedName 'NAME))
      (Extension (LISTGET UnpackedName 'EXTENSION))
      (Directory (LISTGET UnpackedName 'DIRECTORY))
      (Version (LISTGET UnpackedName 'VERSION))
      BadCharPos NewLispFileName (OldLispFileName (ALLOCSTRING (NCHARS LispFileName)))
      (GoodFileName T)
      DirectoryEnd)
  (if (OR (STRING-EQUAL Host "Core")
        (STRPOS ":" Host)
        (STRPOS "dsk" (L-CASE Host)
                1)
        (STRPOS "/n" Host)
        (STRING-EQUAL Host "Floppy"))
    then ;; we can only archive files on an IFS (as of June/88)
      (FB.PROMPTWPRINT BROWSER T (CONCAT "You can only archive files stored on an IFS. The host "
                                          Host " is NOT an IFS!"))
      (SETQ GoodFileName NIL)
    elseif (AND (STRING-EQUAL Name "")
               (STRING-EQUAL Extension ""))
      then ;; This is a file with no name - we want to skip the dif files
        (SETQ GoodFileName NIL)
      elseif (AND (STRING-EQUAL (L-CASE Name)
                               "archive")
                 (STRING-EQUAL (L-CASE Extension)
                               "directory"))
      then ;; we don't want to ask to archive the file archive.directory so just skip it
        (SETQ GoodFileName NIL)
      elseif (STRPOS ":" LispFileName 1)
      then (FB.PROMPTWPRINT BROWSER T (CONCAT LispFileName " has a colon in the name which Lisp can't
                                          handle.")
          " You need to manually call RENAMEFILE to eliminate the colons quoting all colons with
          a single quote.")
          (SETQ GoodFileName NIL)
      elseif (SETQ BadCharPos (STRPOS OKCedarCharBitTable Name 1 T))
      then (FB.PROMPTWPRINT BROWSER T "Invalid character %" (NTHCHAR Name BadCharPos)
          "%" in file " LispFileName ". Only alphanumeric and .$+ are allowed.")
          (if (STRING-EQUAL (FB.PROMPTFORINPUT "Shall I replace all bad chars with $ (Y/N):" "Y" BROWSER
                                          NIL T)
                          "y")
              then (SETQ OldLispFileName (CONCAT LispFileName))
                  (while (AND BadCharPos (ILEQ BadCharPos (NCHARS Name)))
                      do (SETQ Name (RPLCHARCODE Name BadCharPos (CHARCODE $)))
                          (SETQ BadCharPos (STRPOS OKCedarCharBitTable Name BadCharPos T)))
                  (SETQ NewLispFileName (PACKFILENAME.STRING 'NAME Name 'BODY LispFileName))
                  (FB.COPY/RENAME.ONE BROWSER FileEntry OldLispFileName NewLispFileName 'Rename
                                      (FUNCTION RENAMEFILE))
                  else (FB.PROMPTWPRINT BROWSER "Skipping " LispFileName ".")
                      (SETQ SkippedBadFileName T)
                      (SETQ GoodFileName NIL)))
          (if GoodFileName
              then (L-CASE (CONCAT "[" Host "]"<" Directory ">" Name (if (NOT (STRING-EQUAL Extension ""))
                                                                      then (CONCAT "." Extension)
                                                                      else ""))
                          "!" Version]))

```

(FB.ArchiveCommand

[LAMBDA (BROWSER)

(* N.H.Briggs " 3-Apr-86 20:43")

(* Called from FileBrowser Archive command -- Archive all selected files)

(FB.DoArchiveCommands BROWSER 'Archive])

(FB.ArchiveAllCommand

[LAMBDA (BROWSER)

; Edited 28-Apr-88 17:58 by bbb

(* Called from FileBrowser Archive All Files command -- Archive all files)

(FB.DoArchiveCommands BROWSER 'ArchiveAll])

(FB.ArchiveAndDeleteCommand

[LAMBDA (BROWSER)

; Edited 28-Apr-88 15:47 by bbb

(* * Called from FileBrowser Archive and Delete command -- Archive and Delete all selected files)

(**FB.DoArchiveCommands** BROWSER 'ArchiveAndDelete])

(FB.ArchiveAndDeleteAllCommand

[LAMBDA (BROWSER)

; Edited 28-Apr-88 16:52 by bbb

(* * Called from FileBrowser Archive and Delete All Files command -- Archive and Delete all files)

(**FB.DoArchiveCommands** BROWSER 'ArchiveAndDeleteAll])

(FB.DoArchiveCommands

[LAMBDA (BROWSER TypeOfArchive)

; Edited 10-Mar-89 17:03 by bbb

::: Called from FileBrowser Archive command -- Archive all selected files

```
(DECLARE (GLOBALVARS Arch.CcToSelfFlg))
(FB.ALLOW.ABORT BROWSER)
(PROG (FileEntriesList REGISTRY CoreStream CedarFileNameList Pattern MakeRequestFLG FilesSelected
      ArchivedFiles ArchiveDirectory DotStarPosition SkippedBadFileName)
  (if (OR (EQ TypeOfArchive 'Archive)
         (EQ TypeOfArchive 'ArchiveAndDelete))
    then (SETQ FilesSelected T))
  (SETQ Pattern (Arch.ConvertToCedarFileName NIL (fetch PATTERN of BROWSER)
              BROWSER))
```

:: Now we need to fix the pattern because in Cedar *.* will only match files that do have extensions. If there is a "." then replace this with ""

```
if (NULL Pattern)
  then (RETURN)
elseif (SETQ DotStarPosition (STRPOS "." Pattern))
  then (SETQ Pattern (CONCAT (SUBSTRING Pattern 1 (SUB1 DotStarPosition))
                          (if (NEQ (NTHCHAR Pattern (SUB1 DotStarPosition))
                              '*')
                            then ""
                            else ""))
      (SUBSTRING Pattern (PLUS 2 DotStarPosition))
  (SETQ FileEntriesList (if FilesSelected
    then (FB.SELECTEDFILES BROWSER)
    else (TB.COLLECT.ITEMS (fetch (FILEBROWSER TABLEBROWSER) of BROWSER)
      (FUNCTION (LAMBDA (BROWSER ITEM)
        (NOT (fetch TIUNSELECTABLE of ITEM))
      )
    )
  (if (NULL FileEntriesList)
    then (RETURN))
  (if (AND Arch.VerifyNotArchivedFlg [SETQ ArchiveDirectory (PACKFILENAME.STRING 'NAME "Archive"
    'EXTENSION "Directory" 'VERSION NIL
    'BODY
    (SUBSTRING Pattern 1
      (STRPOS ">" Pattern 1]
    (INFILEP ArchiveDirectory))
  then (SETQ ArchivedFiles (CL:MAKE-HASH-TABLE :TEST 'CL:EQUAL))
  ::: Note that the Archive.Directory has: Filename <1 or more spaces> CreationDate <2 spaces>
  (LET ((ArchiveDirectoryStream (OPENSTREAM ArchiveDirectory 'INPUT))
    Line EOF)
    (while (AND (SETQ Line (CL:READ-LINE ArchiveDirectoryStream EOF))
      (NULL EOF))
      bind End FileName CreationDate do (SETQ End (STRPOS " " Line))
      [SETQ FileName (CONCAT (L-CASE (SUBSTRING
        Line 1 (SUB1 End)
      (SETQ CreationDate
        (IDATE (SUBSTRING Line (ADD1 End)
          (SUB1 (STRPOS " " Line (ADD1 End)
        (CL:SETF (CL:GETHASH FileName ArchivedFiles)
          CreationDate))
        (* COLLECTWORDFILE ArchiveDirectory
        (FUNCTION (LAMBDA (Line) (LET*
          ((End (STRPOS " " Line)) (FileName
            (CONCAT (L-CASE (SUBSTRING Line 1
              (SUB1 End)))) (CreationDate (IDATE
                (SUBSTRING Line (ADD1 End) (SUB1
                  (STRPOS " " Line (ADD1 End))))))
          (CL:SETF (CL:GETHASH FileName ArchivedFiles)
            CreationDate)))] NIL (FUNCTION DREADLINE) T)
    )
  )
```

::: Determine right away if they can send mail to the Archivist

```
[SETQ REGISTRY (SELECTQ (OR (LAFITEMODE)
  (\LAFITE.INFER.MODE))
  (GV ".pa"))
```

```

(NS ":PA")
(RETURN (FB.PROMPTWPRINT BROWSER T "Can't -- Lafite mode must be set to GV or NS"])
(if FilesSelected
  then (FB.PROMPTWPRINT BROWSER "Validating and preparing archive request for "
    (CONCAT (LENGTH FileEntriesList)
      " file"
      (COND
        ((CDR FileEntriesList)
          "s")
        (T "")))
    "."))

```

;;; Convert the file names and enter them into the msg. SkippedBadFileName is set to T in Arch.ConvertToCedarFileName if there is an invalid Cedar
 ;;; file which the user chooses not to rename to a valid name

```

(SETQ SkippedBadFileName NIL)
(SETQ CedarFileNameList
  (for FileName in (bind CedarFileName ArchivedFileDate for FileEntry in FileEntriesList
    when (AND (SETQ CedarFileName (Arch.ConvertToCedarFileName FileEntry (
      FB.FETCHFILENAME
      FileEntry)
      BROWSER))
      (if (NULL ArchivedFiles)
        then T
        else (SETQ ArchivedFileDate (CL:GETHASH CedarFileName ArchivedFiles))
          (if (EQ ArchivedFileDate (GETFILEINFO (FB.FETCHFILENAME FileEntry)
            'ICREATIONDATE))
            then (FB.PROMPTWPRINT BROWSER T (FB.FETCHFILENAME FileEntry)
              " has already been archived!" " Skipping it.")
              (SETQ SkippedBadFileName T)
              NIL
            else T)))
          collect CedarFileName)
    collect FileName))
(if CedarFileNameList
  then [if (AND (NOT FilesSelected)
    SkippedBadFileName)
    then
      ;; the person has asked to archive and delete all the files in the browser yet something was wrong with some of their
      ;; names or else it has already been archived
      (SETQ TypeOfArchive (SELECTQ TypeOfArchive
        (ArchiveAll 'Archive)
        (ArchiveAndDeleteAll
          'ArchiveAndDelete)
        (SHOULDNT])

```

;;; Setup the header fields for the msg

```

(SETQ CoreStream (OPENSTREAM ' {NODIRCORE} ' BOTH))
(LINELENGTH 1000 CoreStream) ; In case of long file names

```

;;; Old code (printout CoreStream "Subject: Archive request" T "To: Archivist" REGISTRY T "Cc: ")

```

(printout CoreStream "Subject: Archive request" (SELECTQ TypeOfArchive
  ((ArchiveAll ArchiveAndDeleteAll)
  (CONCAT " for all files in " Pattern
    )
  (CONCAT " for some files in " Pattern))
  T "To: Archivist" REGISTRY T "Cc: ")
[COND
  (Arch.CcToSelfFlg (printout CoreStream (FULLUSERNAME)
  (TERPRI CoreStream)
  (TERPRI CoreStream)
  (SELECTQ TypeOfArchive
    (ArchiveAll (printout CoreStream "Archive: " Pattern)
      (SETQ MakeRequestFLG T))
    (ArchiveAndDeleteAll
      (printout CoreStream "ArchiveAndDelete: " Pattern)
      (SETQ MakeRequestFLG T))
    ((Archive ArchiveAndDelete)
      (for CedarFileName in CedarFileNameList when CedarFileName
        do (SELECTQ TypeOfArchive
          (Archive (printout CoreStream "Archive: " CedarFileName T))
          (ArchiveAndDelete
            (printout CoreStream "ArchiveAndDelete: " CedarFileName T))
          (SHOULDNT))
          (SETQ MakeRequestFLG T)))
      (SHOULDNT))

```

;;; Send the mail off thru the Grapevine

```

(if MakeRequestFLG
  then (SETQ CoreStream (OPENTEXTSTREAM CoreStream NIL NIL NIL (LIST 'FONT LAFITEEDITORFONT)
    ))
  (ADD.PROCESS (LIST (FUNCTION \SENDMESSAGE)
    (KWOTE CoreStream)
    ' NAME

```

```

        'ArchiveRequest])
)
(DECLARE%: DOEVAL@COMPILE DONTCOPY
(GLOBALVARS Arch.CcToSelfFlg Arch.VerifyNotArchivedFlg OKCedarCharBitTable)
)
(RPAQ OKCedarCharBitTable
  [MAKEBITTABLE (APPEND (CHARCODE (%. $ - + * < > { })))
    (for i from (CHARCODE a)
      to
        (CHARCODE z)
      collect i)
    (for i from (CHARCODE A)
      to
        (CHARCODE Z)
      collect i)
    (CHARCODE (0 1 2 3 4 5 6 7 8 9]))
)
(RPAQ? Arch.CcToSelfFlg T)
(RPAQ? Arch.VerifyNotArchivedFlg NIL)
(DECLARE%: DONTEVAL@LOAD DOCOPY
(ADDTOVAR BackgroundMenuCommands (ArchiveBrowser ' (ARCHIVEBROWSER)
  "Brings up an archive browser"))
(SETQ BackgroundMenu)
)
(DECLARE%: EVAL@COMPILE DONTCOPY
(FILESLoad (SOURCE)
  TABLEBROWSERDECLS)
(FILESLoad (LOADCOMP)
  FILEBROWSER)
)
(DECLARE%: DONTEVAL@LOAD DOCOPY
[if (NOT (SASSOC "Archive" FB.MENU.ITEMS))
  then (PUTASSOC "Archive" ' (FB.ArchiveCommand "Archives selected files by sending mail to Archivist.pa"
    (SUBITEMS ("Archive" FB.ArchiveCommand "Archives selected files by sending
      mail to Archivist.pa")
      ("Archive ALL Files" FB.ArchiveAllCommand "Archives ALL files in the
        browser by sending mail to Archivist.pa")
      ("Archive and Delete" FB.ArchiveAndDeleteCommand "Archives selected
        files by sending an Archive and Delete request to
        Archivist.pa")
      ("Archive and Delete ALL Files" FB.ArchiveAndDeleteAllCommand
        "Archives ALL files in the browser by sending an Archive and
        Delete request to Archivist.pa"))
    FB.MENU.ITEMS)
  else (RPLACD (SASSOC "Archive" FB.MENU.ITEMS)
    ' (FB.ArchiveCommand "Archives selected files by sending mail to Archivist.pa"
      (SUBITEMS ("Archive" FB.ArchiveCommand "Archives selected files by sending mail to
        Archivist.pa")
          ("Archive ALL Files" FB.ArchiveAllCommand "Archives ALL files in the browser by
            sending mail to Archivist.pa")
          ("Archive and Delete" FB.ArchiveAndDeleteCommand "Archives selected files by sending
            an Archive and Delete request to Archivist.pa")
          ("Archive and Delete ALL Files" FB.ArchiveAndDeleteAllCommand "Archives ALL files in
            the browser by sending an Archive and Delete request to Archivist.pa"]
      )
)
)
(* * Functions that aren't used any more)
(DEFINEQ
(Arch.ParseReturnMsg
  [LAMBDA (Window)
    (* * Parse an archive response msg contained in Window. Returns a list of the files archived.)
    (PROG (TextObj TextStream CHAR#)
      (RETURN (COND
        ((WINDOWP Window)
          (SETQ TextObj (WINDOWPROP Window 'TEXTOBJ))
          (SETQ TextStream (WINDOWPROP Window 'TEXTSTREAM))
          (SETQ CHAR# 0)
          (while (SETQ CHAR# (TEDIT.FIND TextStream "Archived:" (ADD1 CHAR#)))
            collect (PACKFILENAME 'BODY (U-CASE (PACKC (DSUBLIS

```

```

(CHARCODE (([" . {"}
            "]" . "}"))
(PROGN (SETFILEPTR TextStream CHAR#)
      (READ TextStream)
      (until (NEQ (PEEK TextStream)
                  '% )
             (do (BIN TextStream)
                 (until (EQ (PEEK TextStream)
                             '% )
                       (collect (BIN TextStream))

```

(Arch.BackgroundMenuFn

[LAMBDA NIL

(* fgh%: " 6-Feb-85 01:03")

(* * Archive tool called from background menu. Get from the user a window containing an archive system response message, parse the message, and delete the archived files.)

```

(PROG (Window PromptWindow FileList)
      (PROMPTPRINT "Click in the window containing the response from Archivist.pa")
      (SETQ Window (WHICHW (GETPOSITION)))
      (CLRSPROMPT)
      (COND
        ([AND (WINDOWP Window)
              (TEXTSTREAMP (WINDOWPROP Window 'TEXTSTREAM])
              (SETQ FileList (Arch.ParseReturnMsg Window))
              (COND
                [FileList (SETQ PromptWindow (GETPROMPTWINDOW Window 5))
                          (COND
                            ((MEMBER (PROMPTFORWARD "Okay to delete files? " "Yes" NIL PromptWindow)
                                      '("Yes" "yes" "Y" "y"))
                             (CLEARW PromptWindow)
                             (bind Deleted? for File in FileList do (SETQ Deleted? (DELFILE File))
                               (printout PromptWindow (CONCAT File
                                                                (COND
                                                                  (Deleted?
                                                                    " deleted.")
                                                                  (T " not
                                                                    deleted."))
                                                                T)
                               (DISMISS 500))
                             (printout PromptWindow "Deletions Completed" T)
                             (DISMISS 2000)
                             (REMOVEPROMPTWINDOW Window]
                (T (SETQ PromptWindow (GETPROMPTWINDOW Window 1))
                  (printout PromptWindow "No archived files found in message.")
                  (DISMISS 2000)
                  (REMOVEPROMPTWINDOW Window])
      )

```

```

)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS
(ADDTOVAR NLAMA AB)
(ADDTOVAR NLAML )
(ADDTOVAR LAMA )
)
(PUTPROPS ARCHIVETOOL COPYRIGHT ("Xerox Corporation" 1985 1986 1988 1989))

```

FUNCTION INDEX

AB	4	AB.Read.Directory	9	AB.When.Selected.Fn	4
AB.Closefn	8	AB.Recompute.Command	5	Arch.BackgroundMenuFn	15
AB.Command.Fn	4	AB.Retrieve.Command	5	Arch.ConvertToCedarFileName	11
AB.Delete.Command	4	AB.Retrieve.Directory.Command	6	Arch.ParseReturnMsg	14
AB.Expunge.Command	4	AB.Retrieve.Renamed.Aux	7	ARCHIVEBROWSER	2
AB.Filter.Command	5	AB.Retrieve.Renamed.Command	7	FB.ArchiveAllCommand	11
AB.Iconfn	8	AB.Set.Browser.Title	8	FB.ArchiveAndDeleteAllCommand	12
AB.Make.Cedar.FileName	10	AB.Sort.Command	7	FB.ArchiveAndDeleteCommand	12
AB.Printfn	8	AB.Subitemp	10	FB.ArchiveCommand	11
AB.Prompt.For.Input	9	AB.Undelete.Command	8	FB.DoArchiveCommands	12

VARIABLE INDEX

AB.archivist	10	AB.icon.mask	10	Arch.VerifyNotArchivedFlg	14
AB.browser.font	10	AB.titled.icon	10	BackgroundMenuCommands	14
AB.icon	10	Arch.CcToSelfFlg	14	OKCedarCharBitTable	14

RECORD INDEX

AB.item	10
---------------	----
