F.G.H.
3/28/85

LispCourse #17: Files and Directories ž Part 3

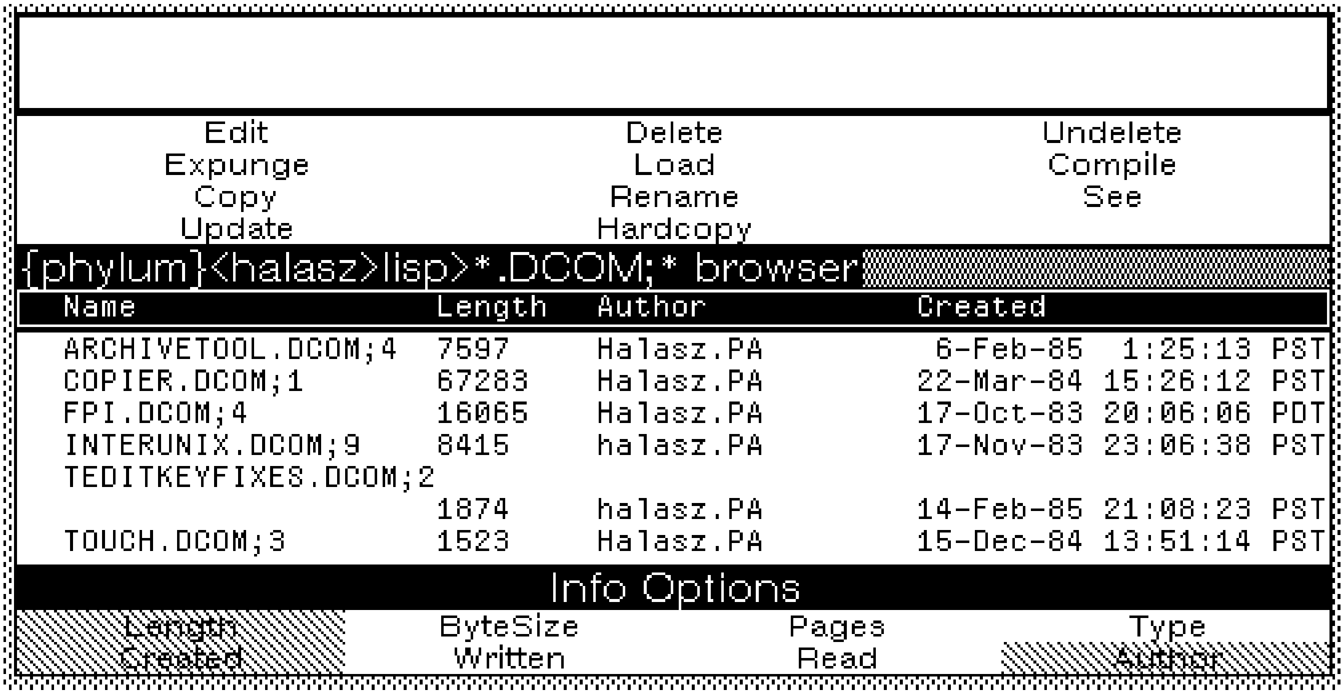
The FileBrowser Package

The FileBrowser is a window/mouse-based interface to the manipulation of files.

To start a file browser, use the function call: (FB *FileNamePattern*) where *FileNamePattern* is a wildcarded file name of the form allowed by FILDIR and DIR (See LispCourse #16, p. 6).

Example: *(FB '{phylum}<halasz>lisp>*.dcom)* will open a file browser on all of the DCOM files in the {phylum}<halasz>lisp> directory. FB will first prompt you for the region in which to place the window.

The result:



```
   Name                 Length   Author          Created
   ARCHIVETOOL.DCOM;4   7597     Halasz.PA        6-Feb-85  1:25:13 PST
   COPIER.DCOM;1        67283    Halasz.PA       22-Mar-84 15:26:12 PST
   FPI.DCOM;4           16065    Halasz.PA       17-Oct-83 20:06:06 PDT
   INTERUNIX.DCOM;9     8415     halasz.PA       17-Nov-83 23:06:38 PST
   TEDITKEYFIXES.DCOM;2
                        1874     halasz.PA       14-Feb-85 21:08:23 PST
   TOUCH.DCOM;3         1523     Halasz.PA       15-Dec-84 13:51:14 PST
```

The FileBrowser display consists of the following 5 divisions from top to bottom:

1) Prompt window ž the place where the FileBrowser prints messages and gets input from you.

2) Command menu ž these are commands that can be applied to the selected file(s).

3) Title bar ž shows the FileNamePattern for this FileBrowser.

4) Files listing ž a scrollable listing of the names and attributes of files matching FileNamePattern.

5) Info Options menu ž a menu of the file attributes to be displayed in the Files listing.

The Files Listing and Selecting Files

The files listing can be scrolled like any window; just roll out the left side of a window to get the scroll bar.

One or more files in the files listing can be selected at any given time. To select a file, point to the left end of any line and click a mouse button. When the cursor nears the left edge of the window, it turns into a right pointing arrow (-->).

The mouse buttons are interpreted as follows:

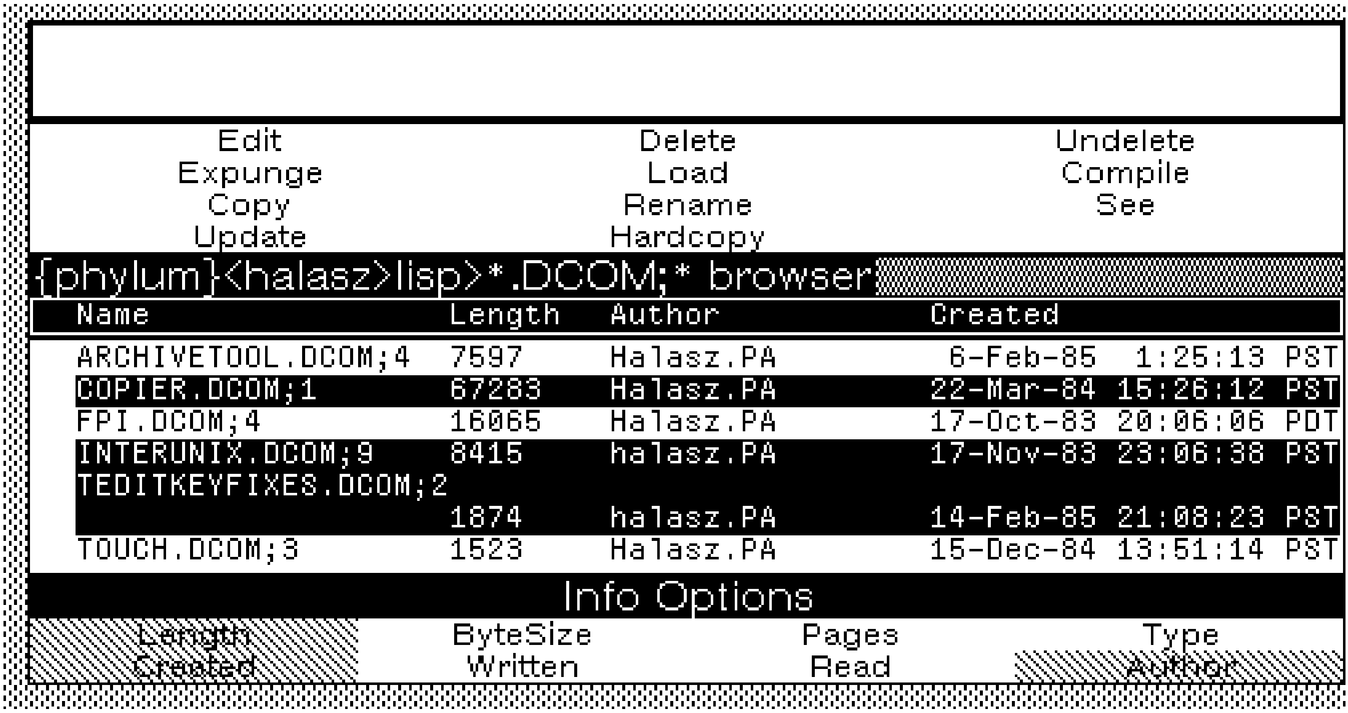
LEFT ž selects the file and unselects all other files (i.e., makes the file the ONLY selection).

MIDDLE ž inverts the selection of the file, i.e., if the file was selected then it unselects the file otherwise it selects the file. All other selections are unaffected. (This button is used for selecting several files that are not contiguous.)

RIGHT ž extends a selection, i.e., selects the file and all files between this file and the nearest selected file. Exceptions: Has no affect if the file is already selected. Has no affect if the chosen file is BEWTEEN two already selected files.

Note: The LEFT and MIDDLE buttons can actually be clicked when the cursor is anywhere in the line. The RIGHT button must be clicked when the cursor is at the left edge of the line.

Selecting a file inverts the line:



{phylum}<halasz>lisp>*.DCOM;* browser

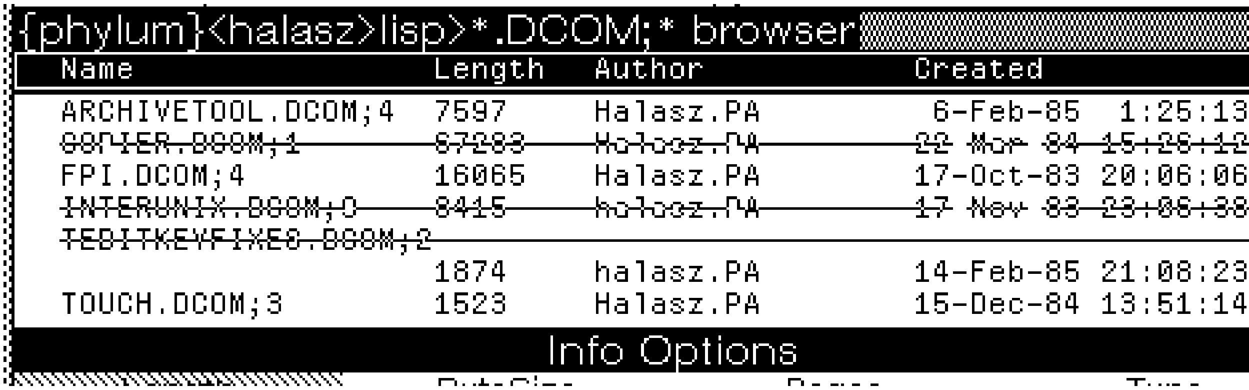
```
Name                  Length    Author            Created
ARCHIVETOOL.DCOM;4    7597      Halasz.PA          6-Feb-85  1:25:13 PST
COPIER.DCOM;1         67283     Halasz.PA         22-Mar-84 15:26:12 PST
FPI.DCOM;4            16065     Halasz.PA         17-Oct-83 20:06:06 PDT
INTERUNIX.DCOM;9      8415      halasz.PA         17-Nov-83 23:06:38 PST
TEDITKEYFIXES.DCOM;2
                      1874      halasz.PA         14-Feb-85 21:08:23 PST
TOUCH.DCOM;3          1523      Halasz.PA         15-Dec-84 13:51:14 PST
```

The Command Menu

Most of the commands in the command menu carry out an operation on the selected files. The commands work as follows:

Edit ž Starts up an editor (TEdit or DEdit) on each of the selected files. If the command is selected using the LEFT mouse button, DEdit is called on Lisp symbolic files and TEdit on all other files. If the command is selected using the MIDDLE mouse button, you will be asked to choose the editor to use from a menu.

Delete ž Marks the selected files for deletion but does not actually delete them. An expunge command is required to do the actually deletion. Files marked for deletion have a line drawn thru them.



```
{phylum}<halasz>lisp>*.DCOM;* browser
   Name                 Length    Author                 Created
   ARCHIVETOOL.DCOM;4   7597      Halasz.PA               6-Feb-85  1:25:13
   ~~COPIER.DCOM;1       67283     Halasz.PA              22-Mar-84 15:26:12~~
   FPI.DCOM;4           16065     Halasz.PA              17-Oct-83 20:06:06
   ~~INTERUNIX.DCOM;0    8415      halasz.PA              17-Nov-83 23:06:38~~
   ~~TEDITKEYFIXES.DCOM;2~~
                        1874      halasz.PA              14-Feb-85 21:08:23
   TOUCH.DCOM;3         1523      Halasz.PA              15-Dec-84 13:51:14
                              Info Options
```

Undelete ž Removes the deletion marks (if any) from the selected files so that they will not be deleted in the next expunge.

Expunge ž Actually deletes all of the files that have been marked for deletion and them updates the files listing. Not this command ignores the selected files.   It operates on the *marked* files instead.

Load ž Applies LOAD to the selected files.

Compile ž Compiles the selected files.

Copy ž Copies the selected files using a COPYFILE for each file. You will be prompted for the destination of the COPYFILEs in the FileBrowser's prompt window. If there is a single selected file, you will be asked for a destination *file name*. If there is more than one file selected, you will be asked for a destination *path name* (i.e., a directory). In the latter case, each of the selected files will be copied to the specified directory retaining their old file name.

Rename ž Like Copy, except does a RENAMEFILE for each selected file.

See ž For each selected file, opens a separate window (prompting for a region) and prints the contents of the file in the window. The windows are scrollable but not editable.

Update ž Updates the files listing. Updates are necessary if a change is made to the listed directory. Except following the Expunge command, the FileBrowser does not automatically update the files listing when changes are made to the files.

The Update command must be used to insure that the file listing reflects the true state of the files listed.

If this command is selected using the LEFT mouse button, a file listing is made using the current FileNamePattern. If the MIDDLE mouse button is used, you are given the a choice of using the current FileNamePattern or specifying a new FileNamePattern for the file listing. If the new FileNamePattern option is chosen, you will be promptyed for the new pattern in the prompt window.

Hardcopy ž Applies LISTFILES (See LispCourse #16, p. 13) to the selected files.

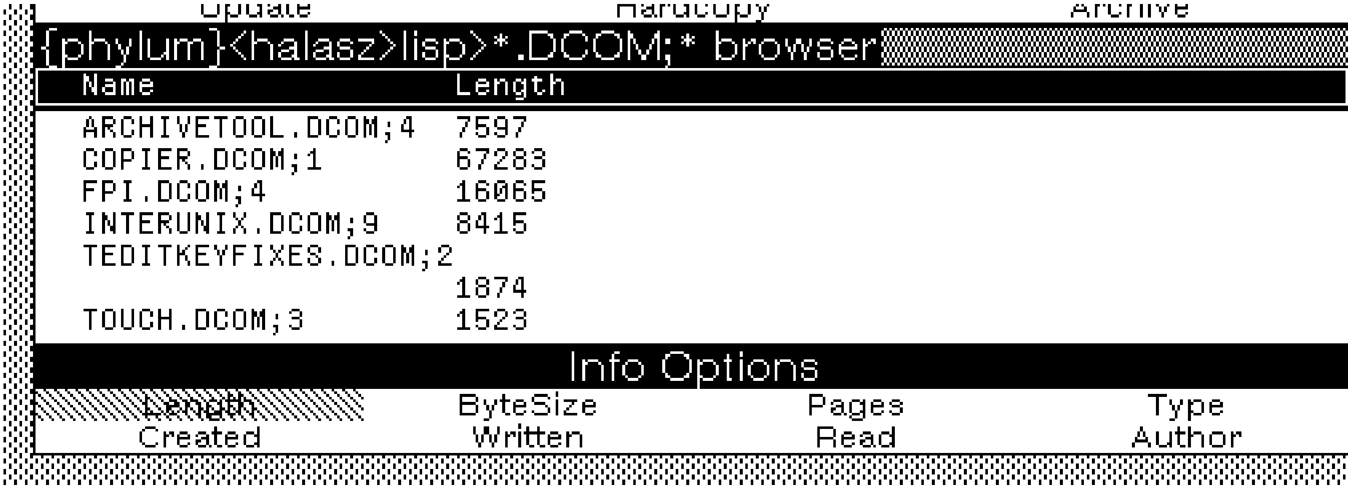
The Info Options Menu

The Info Options menu can be used to set what attributes of a file get displayed in the file listing. Each item in the menu represents one of the attribute of a file (e.g., Length, CreationDate, Author, etc.).

Clicking the mouse in the item inverts the selectedness of the item. Selected items are shaded.

When the file listing is updated (e.g., using the Update command) the line for each file will include the selected attributes.

Example: Only Length attribute selected.



{phylum}<halasz>lisp>*.DCOM;* browser

```
Name                Length
ARCHIVETOOL.DCOM;4  7597
COPIER.DCOM;1       67283
FPI.DCOM;4          16065
INTERUNIX.DCOM;9    8415
TEDITKEYFIXES.DCOM;2
                    1874
TOUCH.DCOM;3        1523
```

Using the FileBrowser

You can simultaneously open as many FileBrowsers as you like.

Beware of the order in which files are listed in the FileBrowser. Each device appears in a different order. In particular, you should be careful to observe the order in which the several versions of a file are listed. On some devices, versions will be in ascending order, on other devices in descending order, on still other devices in lexical or even random order.

Dealing with Devices

In general, you deal with files and directories and NOT directly with the devices these files and directories are stored. However, there are several properties of devices that are important to the user:

1. Amount of available space on the device

Each device can store only a limited amount of data. Often you want to know how much of this space is left. The following functions provide that information for various devices:

Local Disk ž the function (DISKFREEPAGES *DeviceName*) returns the number of pages (512 bytes) still availabel on the named local disk partition**.**

Example:

```
77_(DISKFREEPAGES ’DSK5)
8341
78_
```

File Servers ž There is no Interlisp interface to finding out the amount of space left on a directory on a file server. For the IFSs, you can CHAT to the IFS and then CONNECT to the directory and issue the DSKSTAT command.

Example:

```
Phylum IFS 1.38.1L Executive ...
@login (user) halasz (password)
@connect (to directory) notecards
@dskstat
35373 pages used out of 40000 in directory <notecards>.
495072 pages used, 28872 left in the system.
@quit
```

Floppy ž To find out the amount of space left on a floppy disk, use the (FLOPPY.FREE.PAGES) function call.

Example:

```
79_(FLOPPY.FREE.PAGES)
345
80_
```

Core Devices ž The concept of space is not relevant here since these are "simulated" devices. In general, there is only about 5000 pages available for COREDEVICE files in an Interlisp virtualmem.

2. Random access versus No random access

Some device allow programs to access the contents of a file in a random order. Other devices allow access to a file in a strict sequential order from start to finish.

Some programs require random access devices. Lafite, in particular, require mail files to be stored on random access devices.

All devices we have discussed allow random access EXCEPT the NS file servers. Therefore many programs including Lafite cannot be used with NS file servers.

3. Location of your virtual memory file

The function call (DISKPARTITION) returns the number (on Dolphin/Dorado) or name (on DLion) of the local disk partition from which the currently running Lisp has been started, i.e., the partition with the currently active virtual memory file.

On Dolphin/Dorados, this will also be the partition number of the {DSK} partition.

Using Floppies on the Dlion

On the whole, floppies on the DLion can be treated like any other device. You can COPYFILE to a from a floppy, RENAMEFILE files on a floppy, etc.

There are some special procedures for dealing with floppies.

The Physical Floppy

Insertion

Write Enable Tab

Floppy Modes

There are three modes for writing on floppies. The modes are defined as follows:

PILOT ž This is the normal mode. In PILOT mode, each floppy can contain many files. No file can be bigger than the amount of information that can fit on one floppy (about 2200 pages).

HUGEPILOT ž In HUGEPILOT mode, there is exactly one file per floppy AND each file can spread across several floppies. HUGEPILOT is generally used for files that are bigger than a single floppy.

SYSOUT ž SYSOUT mode is like HUGEPILOT mode. It is used for Lisp sysouts, which are always large files that take up more than one floppy.

Note: That the floppy mode is a property of each floppy in the sense that any given floppy can be written on in only one of these modes. You cannot write on a floppy in HUGEPILOT mode once and then PILOT mode the next time.

(FLOPPY.MODE *Mode*) is the function to change the current floppy mode. *Mode* must be one of the three modes described above. The value returned is the previous floppy mode. (FLOPPY.MODE) just returns the current floppy mode.

When reading from a floppy (e.g., COPYFILEing from a floppy to another device), the system will detect what mode is appropriate a read the file from the floppy in the correct mode.

When writing to a floppy (e.g., COPYFILEing from a file server to a floppy), you must set the floppy mode as appropriate. If you are in HUGEPILOT or SYSOUT mode, the system will prompt you for another floppy whenever it needs one.

Example: Copying a sysout from phylum to floppy.

```
8_ (FLOPPY.MODE ’PILOT)
PILOT
9_  (COPYFILE ’{PHYLUM}<halasz>Init ’{FLOPPY}INIT)
{FLOPPY}INIT;1
10_ (FLOPPY.MODE ’SYSOUT)
```

```
PILOT
11_  (COPYFILE '{phylum}<halasz>new.sysout
'{floppy}lisp.sysout)
Please insert floppy for Lisp Sysout #1
<about 15 min passes>
Please insert floppy for Lisp Sysout #2
<about 15 min passes>
Please insert floppy for Lisp Sysout #3
{floppy}lisp.sysout;1
12_ (FLOPPY.MODE 'PILOT)
SYSOUT
```

Formatting, Compacting, Scavenging

(FLOPPY.FORMAT *Name*) ž Before being used, every new floppy must be formatted. Formatting gives the floppy a name and sets up the floppy so Interlisp can write on it. The *Name* of a floppy is any arbitrary atom.

> Formatting a floppy erases the information already on the floppy which can be good or bad, depending on whether you intended to erase the information or not.

(FLOPPY.COMPACT) ž files on a floppy need to take up contiguous space. Sometimes you may have enough free space on your floppy for a file, but the file still won't fit because the free space is not contiguous but spread out between existing files. (FLOPPY.COMPACT) will compact the free space on a floppy so that all free space is contiguous.

> For example, if you create three files and then delete the middle one, you will have some discontiguous free space (i.e., the space previously taken up by the deleted file between the two existing files and the free space after the third file). The space between the two files cannot be used by large files. (FLOPPY.COMPACT) will move the last file so that all the free space is in one block at the end of the floppy, thus making it possible for all of the free space to be used by large files.

(FLOPPY.SCAVENGE) ž tries to fix up a floppy that has gotten messed up for some reason or another.

> FLOPPY.SCAVENGE will also retrieve files that have been accidentally deleted, providing the space used by those files has not already been overwritten by another file. In particular, if you accidentally delete a file and run FLOPPY.SCAVENGE before writing another files on the disk, you will always get back your deleted files.

Storing and Retrieving Floppy Images

Sometimes its convenient to treat each floppy as one giant file, no matter how many files are actually stored on the floppy. For example, you might want to copy a floppy containing several files to a file server and then make a copy of the floppy from the file server. One procedure would be to do the copies file by file using COPYFILE. A quicker procedure, however, would be to treat the floppies as one giant file and copy that one file back and forth to the file server.

The functions FLOPPY.TO.FILE and FLOPPY.FROM.FILE allow you to treat a whole floppy as a single giant file.

(FLOPPY.TO.FILE *FileName*) ž will copy a whole floppy (treated as a single file) to the file named by *FileName*.

(FLOPPY.FROM.FILE *FileName*) ž will copy the file named by *FileName* to a whole floppy (treated as a single file).

Example:

```
10_ (FLOPPY.TO.FILE '{PHYLUM}<halasz>WHOLEFLOPPY)
{PHYLUM}<halasz>WHOLEFLOPPY;1
<<Insert a blank floppy>>
11_ (FLOPPY.FROM.FILE '{PHYLUM}<halasz>WHOLEFLOPPY)
{PHYLUM}<halasz>WHOLEFLOPPY;1
<<Insert a blank floppy>>
12_ (FLOPPY.FROM.FILE '{PHYLUM}<halasz>WHOLEFLOPPY)
{PHYLUM}<halasz>WHOLEFLOPPY;1
```

Documentation

Can be found on *{eris}<lisp>harmony>doc>floppy.tty*.

Opening and Closing Files

Before a program to read from or write to a file, the file must be opened by Interlisp. The file remains open as long as the program is reading from or writing on the file. When the program is finished, it closes the file.

If one program is writing on an open file, no other program can open that file. The second program will get a "File won't open" error when it tries to access the file.

Two programs can read from the same file at the same time.

Ordinarily, opening and closing of files is done automatically. Sometimes, however, a program bombs without properly closing its files. In this case, the files remain open and can prevent orther programs from accessing the files. When this happens, the user has to close the open files by hand.

The following function are used for closing files:

(OPENP) ž returns a list of all the open files.

(CLOSEALL) ž closes all open files

(CLOSEF *FullFileName*) ž closes the file specified by *FullFileName*.

Example:

```
10_ (TEDIT '{phylum}<halasz>test.ted)
File won't open
{phylum}<halasz>test.ted;1
11_ (OPENP)
({phylum}<halasz>test.ted;1)
12_ (CLOSEALL)
({phylum}<halasz>test.ted;1)
13_ (TEDIT '{phylum}<halasz>test.ted)
{process}#6.23456
14_
```

References

There is no FileBrowser documentation

Floppy documentation is on *{eris}<lisp>harmony>doc>floppy.tty*.

Opening and closing files is covered in Section 6.1 of the IRM