

FILE BROWSER

The File Browser provides a convenient user interface for manipulating the files stored on a workstation or file server. It enables you to see, edit, delete, print, load, copy, move, rename, compile, and get several types of information about groups of files. You can also customize the File Browser by adding your own commands.

Loading the File Browser

To load the File Browser, type (LOAD {DEVICE}<DIRECTORY>FILEBROWSER.DCOM) in your Interlisp-D executive window.

Once you have loaded the File Browser, there are two ways to open a browser on a set of files. You can select the FileBrowser command from the background menu, in which case you are prompted for a file name pattern, or you can type the command FB FILEPATTERN in your executive window. You can simultaneously have as many active File Browsers as you like.

Specifying What Files to Browse

A full file name in Interlisp-D consists of a device or host (such as your local disk, a file server, or a floppy disk), a principal directory and zero or more subdirectories, a file name (possibly including an extension), and a version number. These fields are put together in the form {DEVICE}<DIRECTORY>SUBDIRECTORY>FILENAME.EXTENSION;VERSION. (In this document, capital letters are used for file names for emphasis. However, you may use lowercase letters where you wish; file names are not case-sensitive in Interlisp.)

A file name *pattern*, as specified to the File Browser, consists of a file name with one or more pieces omitted or filled with wild cards. All the files matching the pattern are listed by the File Browser. Thus, you can browse all the files in a particular directory, all the files in a subdirectory of that directory, all the files in a directory with a particular extension, and so forth. The wild card "*" can be used to stand for zero or more consecutive

characters in the file name. You can use as many wild cards in a pattern as you wish.

If you leave out some of the fields in a file name pattern, the missing fields are defaulted by the system. Omitted fields in the front of the pattern, i.e., the host, device, or directory fields, are filled in by consulting your connected directory. Other omitted fields are filled in with wild cards, unless they are "explicitly omitted," i.e., the field is empty, but the preceding punctuation is still present. In more detail, some of the cases are as follows:

If you leave out the name of the host/device, specifying <DIRECTORY>FILENAME, the File Browser will use the name of the host/device for the directory to which you are currently connected.

If you leave out both the device and directory names, specifying FILENAME, the File Browser will use the device and directory to which you are currently connected.

If you do not specify a file name, the File Browser lists all the files in the specified directory (or the connected directory if you also omitted the host and directory).

If you leave out the extension of a file name, the File Browser lists all the files with the specified file name and any extension. If you omit the extension but include the period that usually precedes the extension, the File Browser lists only the files with the specified name and *no* extension.

If you omit the version number of the file name, the File Browser lists all versions of the matching files. If you omit the version number but include the semicolon that usually precedes the version, the File Browser lists only the highest version of the matching files.

Some examples:

The pattern "*" specifies all files in the connected directory. It is equivalent to "*.*" or "*.*;*".

The pattern "<FOO>BAR" specifies all files in directory FOO with name BAR and any extension. It is equivalent to "<FOO>BAR.*;*".

The pattern "<FOO>BAR." specifies all files in directory FOO with name BAR and *no* extension. It is equivalent to "<FOO>BAR;*".

The pattern ".TEdit" specifies all files in the connected directory with the extension "TEdit." It is equivalent to "*.TEdit;*".

The pattern ".TEdit;" specifies only the newest version of all files in the connected directory with the extension "TEdit".

The pattern "<FOO>A*E" specifies all files in directory FOO whose names begin with "A" and end with "E" and have any extension.

The pattern "{TOAST}<FOO>*MY*" specifies all files in directory {TOAST}<FOO> whose names contain the substring "MY" and any extension.

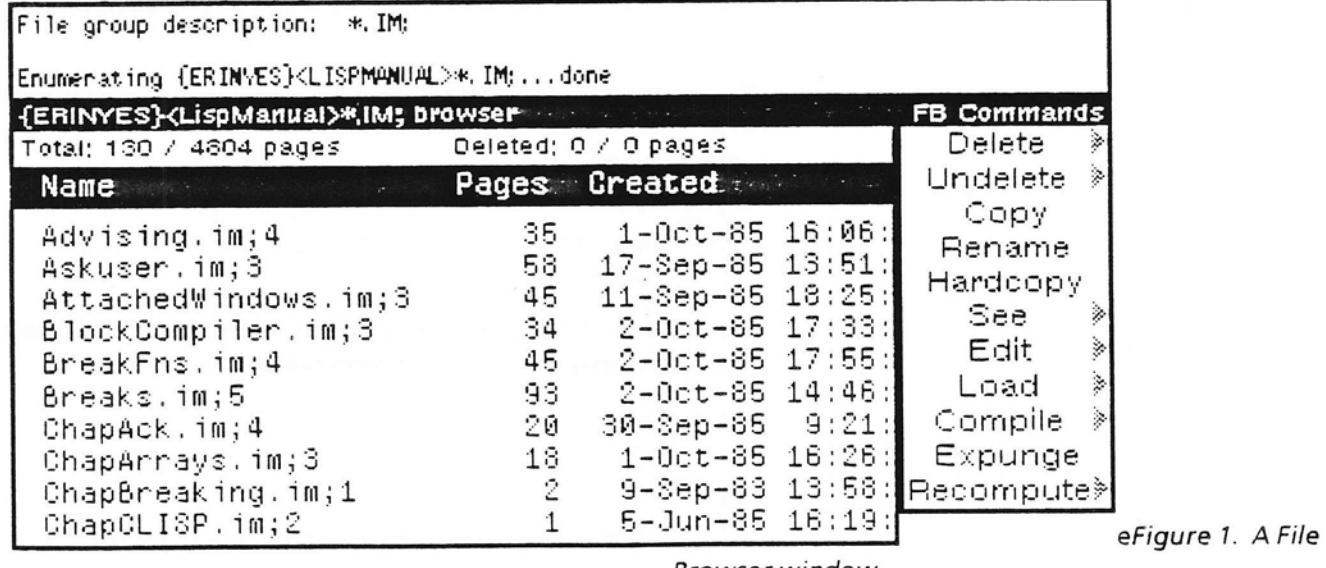
Opening a File Browser Window

After you issue the FB or FileBrowser command, the File Browser will prompt you to create a window by presenting you with a dashed rectangle with the mouse cursor and a small geometric design at the lower-right corner. Move your mouse until the window is where you want it on the screen, then hold down the left mouse button and move your mouse down and to the right, thus expanding the window diagonally, until the window is the size you want. The FB command does no special processing on the characters whatsoever. It simply displays each file in the outlined area.

If you selected the FileBrowser command from the background menu, the File Browser now prompts you for a file pattern. Type a pattern, as described in the previous section. (If you issued the FB command, you have already specified the pattern.)

The File Browser now enumerates the set of files matching the pattern you requested to see. While the enumeration is in progress, the Recompute command is grayed out. When the enumeration is finished, you may scroll the window, select files, and issue commands.

If the File Browser can't find any files matching the pattern you specified, or you decide you specified the wrong pattern and want to try again, you can specify a new file name pattern from within the browser using the New Pattern command; see "Recomputing the File Listing."



eFigure 1. A File

Browser window

Using the File Browser Window

The File Browser window has six major subwindows, which from top to bottom are as follows (see figure 1):

The *prompt* window.

This topmost subwindow is where the File Browser prints messages about what it is doing and receives input from you. Its contents are cleared before every command. If you want to save some screen space, you can close this subwindow using the standard right-button window menu—it will reappear the next time the File Browser needs it.

The *tally* window.

This subwindow immediately below the prompt window keeps a running tally of the total number of files listed in the window and the number of files that you have marked for deletion. In addition, if one of the attributes you are displaying is a size attribute (Pages or Length, see below), this window maintains a tally of the total number of pages in the files listed and the files marked for deletion. This window also has a title bar across the top identifying the pattern you specified. The window is blank while the files are being enumerated.

The *browser* window.

This is the principal subwindow, in which the files matching the specified pattern are listed. Each file's name appears at the left, and various attributes of the file are displayed in columns to the right. A black bar across the top of the browser window identifies the contents of each column. The files are listed in alphabetical order, with multiple versions of the same file listed in decreasing version order; i.e., the newest version appears first. The width of the column listing the file names is initially chosen to be appropriate for average-sized file names. If the files you asked to browse have particularly long names, then when the File Browser has finished listing all the files it may choose to redraw the browser window with the attribute columns moved farther to the right to accommodate the longer file names.

The *command menu*.

This menu appears vertically along the right side of the window, and contains commands that you may select to perform operations on the files in the browser, or change the appearance of the browser. Most of the commands operate on the set of currently selected files (see "Selecting Files"). Some commands have subcommands, as indicated by the small triangle listing them, which can be selected by holding down the left mouse button and rolling the mouse rightwards over the triangle.

The *info menu*.

An additional subwindow, the *info menu*, is not normally displayed. It is used to change the set of file information

(attributes) displayed in the browser (see "Getting Information About Files").

The scroll bar.

If there are more files in the listing than fit at one time in the browser window, you can scroll the browser window to view more files. Roll the mouse cursor out the left side of the window to get the scroll bar and press the left mouse button to scroll the region up and the right mouse button to scroll it down. Clicking a mouse button when the cursor is near the bottom of the scroll bar will scroll the region by larger increments than when the cursor is at the top. You can also press the middle mouse button in the scroll bar to move the listing to the place that corresponds to that position in the scroll bar. For example, pressing the middle mouse button when the cursor is at the bottom of the scroll bar will display the end of the listing. This quick-scrolling technique is called thumbing. The grey box in the scroll region indicates where the currently displayed contents are relative to the entire contents of the browser.

Similarly, if there is more attribute information than fits in the browser window, you can scroll the browser window horizontally to view the rest of the attribute information. To do this, roll the mouse out the bot-

of the window to get the horizontal scroll bar. The left button scrolls to the left, the right button to the right.

Selecting Files

Most File Browser operations are performed by selecting a single file or set of files, then giving a command that specifies what you want to do with the selected files. The current selection is indicated by a small triangle in the left margin of the browser next to each selected file; see figure 2.

{PHYLUM}><vanMelle>LISP>*>* browser				FB Commands
Total: 73 / 804 pages		Deleted: 0 / 0 pages		
Name	Pages	Created		
CPPFILES.;2	10	20-Nov-84 16:30:4		Delete >
CPPFILES.DCOM;2	6	20-Nov-84 16:32:4		Undelete >
►DECODE.;7	32	23-Nov-83 16:46:4		Copy
►DLCLOCK.;2	3	25-Feb-85 16:33:5		Rename
►DLCLOCK.DCOM;2	2	25-Feb-85 16:34:1		Hardcopy
►OPYHACKS.;3	4	21-Jan-83 11:56:5		See >
►OT.;4	20	22-Nov-83 12:37:0		Edit >
EO.;1	8	3-Dec-81 10:48:3		Load >
EE.;1	8	10-Dec-81 17:12:1		Compile >
				Expunge
				Recompute >

selected files

Figure 2. Several

To select one file, point to any part of the line and click the left mouse button. If other files are already selected, this unselects

them; thus, a file selected with the left mouse button is always the only selection.

To add a single file to the current selection, click the middle button. Holding the selection avoids deleted files (see "Selecting Files"). For example, if you wanted to copy only the most recent version of all the files in the browser to aol key and click the middle mouse button at any place in the line. The file is unselected without affecting any other file.

To extend the selection to include a group of contiguous files, that is, to select all the files between a file and the nearest already selected file, click the right mouse button on any part of the line. You can only extend the selection from the first selected file upward, or the last selected file downward. In addition, files marked for deletion are not normally selected when you extend. If you want to include all files, both deleted and undeleted, hold down the control key while extending the selection.

Some lines in a File Browser display are "directory-only" lines. These lines are slightly indented and name the directory and subdirectory to which the files listed below that line belong. You cannot select in these lines.

figure 8). You can

a

cept the default or supply your own destination file name. If you supply just a directory specification, e.g.,

{SERVER}<YOURDIRECTopy (or shift) key while selecting a name in the window. The full name of the file is inserted as if you had typed it where the input caret is flashing. You can also copy-select in a directory-only line, in which case the full directory name is inserted in your type-in.

Issuing File Browser Commands

Some File Browser commands require input from you. For example, the Copy command requires that you supply a destination for the file. When a command requires input, the File Browser prints a prompt message in its prompt window. This is usually followed by a "default answer." If you like the default answer, you can just press the carriage return to finish the input. If you want to specify a different answer, simply start typing it; the default answer is erased and your answer replaces it. Alternatively, you can modify the default answer by backspacing over individual letters, or typing control-W to back up over complete words. Typing control-Q erases the entire answer. In any case, when you are finished, press the carriage return.

If you change your mind and want to abort the command, supply an empty input; i.e., if there is an answer in progress, backspace over it or type control-Q to erase it, then press the carriage return. The File Browser prints "aborted" and aborts the command. In most situations, the control-E interrupt can also be used to abort your answer.

While you are typing an answer, you can copy-select file names out of the browser (or any other browser), as described above. This can be useful, for example, if you wish to rename a file to a similar name in the same directory, or move a file into a subdirectory listed in the browser.

Seeing Files

When you browse a directory you sometimes want to see a file before printing or performing some other operation on it. To do this, select the file, then click on the See command from the command menu. The File Browser will prompt you to open a window by presenting you with a dashed rectangle and printing a message in your Interlisp-D prompt window. The window will be blank until the File Browser starts printing the contents of the file in it.

There are actually three different See commands, as shown in the roll-out submenu for the See command (see figure 3). The two "fast" See commands are provided to let you quickly see the contents of a file, but not do anything fancy, such as scroll around at random in the file. The slower "scrollable" command does let you scroll, and if the file contains formatting information of a kind that the File Browser knows about, you will see the file formatted. However, this command does much more work, and may take a bit longer to show you even the first line of the file. The different versions of See are described further below.

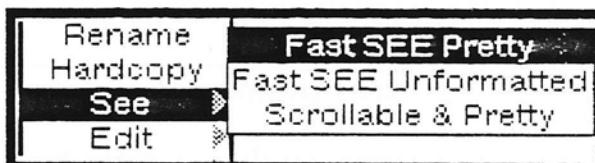


Figure 3. See subcommands

The two fast See commands print the selected file into the display window one windowful at a time. When the file fills the window, a small menu appears at the bottom-left corner of the window (or top-left if your display window is at the bottom of the screen) giving you the option of seeing more of the file or aborting the See command. If you issued the See command with more than one file selected, you also have the choice of aborting just the display of this file or the entire See command (see figure 4).

```

Viewing {ERIS}<Lisp>Koto>Lispusers>ACTIVEREGIONS.;1
(FILECREATED " 6-JUL-84 17:48:34" {SDHDX1}DISKO:<DOLPHIN.LISPUS
ERS>ACTIVEREG.;8 7633
    changes to: (FNS ACTIVEREGIONS/MULTIPLEREGIONS? FINDACTIVEREGIO
N
        ACTIVEREGIONS/DEFAULTHIGHLIGHTFN)
        (VARS ACTIVEREGIONSFNS ACTIVEREGIONSHIDDENFNS)
        previous date: "15-MAR-84 08:43:50" {SDRVX1}DISKO:<DOLPHIN.LISP
USERS>ACTIVEREG.;5) **COMMENT**
(PRETTYCOMPRESS ACTIVEREGIONSCOMS)
(RPAQQ ACTIVEREGIONSCOMS ( **COMMENT**
    (RECORDS ACTIVEREGION)
    (FNS * ACTIVEREGIONSFNS) **COMMENT**
    (FNS * ACTIVEREGIONSHIDDENFNS)))
**COMMENT**
[DECLARE: EVAL@COMPILE
(RECORD ACTIVEREGION (REGION HELPSTRING DOWNFN UPFN HIGHLIGHTFN LO
WLIGHTFN DATA))
]

```

More Next File Abort

Figure 4. A Fast See Formatted display window awaiting the next command

If you click on More, the See command displays another windowful of the file. If you click on Next File, the See command closes this file and goes on to display the next file in the current selection. If you click on Abort, the entire See command is aborted. You can also abort the See command by closing the display window.

The next time you give a Fast See command, the same window will be reused.

The only difference between the Fast See Formatted and the Fast See Unformatted commands is the manner in which the characters of the file are processed as they are displayed.

The Formatted version interprets certain control characters found in Interlisp-D source files to be font change commands, and interprets certain multibyte sequences as representing characters in the Xerox extended character set. It also squeezes out blank lines and shrinks the indentation of indented lines in order to better fit the text in a window that is generally much narrower than the standard file width. The Formatted version is thus most appropriate for viewing Interlisp-D source files and files containing plain text.

The Unformatted version of the See command does no special processing on the characters whatsoever. It simply displays each byte window (for the byte that represents the ASCII "bell" character) and bytes that do not represent normal printing characters may be displayed as black boxes, in the form " $\uparrow x$ " or "#x," or as a flashing of the window (for the byte that represents the ASCII "bell" character). The Unformatted version is thus most appropriate for viewing "binary" files that also contain text

portions that might be worth seeing; e.g., Interlisp-D compiled files (those with extension DCOM) or Interpress masters (e.g. Browser pr or INTERPRESS).

The See Scrollable & Pretty command views a file in a different way. This command brings up a read-only TEdit window viewing the file. You can scroll and copy-select the file's contents at will, as with any TEdit window. If the file is an Interlisp-D source file, its contents are first formatted into a TEdit document, so that all the font information is retained. This formatting, however, can take a long time for a large file. For other kinds of files, the See Scrollable & Pretty command is exactly like viewing the file in a regular TEdit window, except that you can't edit it. If you want to edit a file, use the Edit command instead of the See command.

You can keep the display window used by the See Scrollable & Pretty command open as long as you like. The command uses a different window for each file you select. Simply close the window with the standard right-button window menu when you are finished with it.

Editing Files

To start up a TEdit editor on a selected text file, click on Edit with the left mouse button. If you have recently closed a TEdit window, then TEdit will probably reuse that window; otherwise, you will be prompted to create an editor window. TEdit only file could not be renamed, this is noted in the prompt window. The reasons for the failure of a renaming operation are roughly the same as for the failure of an Expunge!!! the file is open, or you do not have the access rights needed to rename the file.

2

tation. When you are finished with the Edit window, you can simply close the window.

Printing Files

You can print text files, TEdit files, Interpress or Press files, and Lisp files from the File Browser. Select the appropriate file or files, then click on the Hardcopy command. The Hardcopy mation about the files as they are loaded (see figure 10). When the load is complete, the File Browser closes the Load window.

■ are being printed and when they are done (if your printer is one that provides this status service).

For files stored on non-random-access servers, the File Browser is currently unable to determine that a file is in TEdit format unless

the file has the extension .TEdit. Therefore you should use the TEdit Hardcopy command instead.

Deleting and Undeleting Files

Removing a file from the file system using the File Browser is a two-step process. First, you mark the file or files for deletion. Then you issue the Expunge command. Any time between the deletion and the expunge you can change your mind and undelete any of the files.

To mark a file or files for deletion, select them, then choose the Delete command. The File Browser draws a line through the deleted files (see figure 5). It also adjusts the figures in the tally window to show how many files are marked deleted and how many pages they contain. It is thus easy to see how much space you will regain when you issue the Expunge command.

{ERIS}<Lisp>Koto>Lispusers>*.* browser				FB Commands
Total: 744 / 23088 pages		Deleted: 3 / 458 pages		
Name	Pages	Created		
ACTIVEREGIONS.;1	16	10-Dec-84 19:05:5		Delete ➤
►ACTIVEREGIONS.DCOM;1	8	15-Mar-84 4:45:2		Undelete ➤
activeregions.tty;1	7	19-Jan-85 8:07:3		Copy ➤
AIREGIONS.;1	52	24-Jun-85 9:54:5		Rename ➤
AIREGIONS-DEMO.;1	191	24-Jun-85 0:59:1		Hardcopy ➤
AIREGIONS-DEMO.DCOM;1	191	24-Jun-85 9:00:0		See ➤
AIREGIONS.DCOM;1	19	24-Jun-85 9:55:3		Edit ➤
AIREGIONS.IP;1	150	23-Jun-85 18:17:3		Load ➤
AIREGIONS.PREC8;1	76	23-Jun-85 10:19:0		Compile ➤
AIREGIONS.TEDIT;1	64	23-Jun-85 18:16:1		Expunge ➤
ALL.;1	14	20-Dec-82 16:59:1		Recompute ➤

Figure 5. Some files marked for deletion

To undelete a file or files (i.e., to remove the deletion mark), select them, then choose the Undelete command. The lines through the files are removed, and the tally of deleted files is updated. The Undelete command has a single subcommand, Undelete All Files (see figure 6), which undeltes all the files in the browser, independently of whether they are selected. This is useful if you completely change your mind about deleting any files.

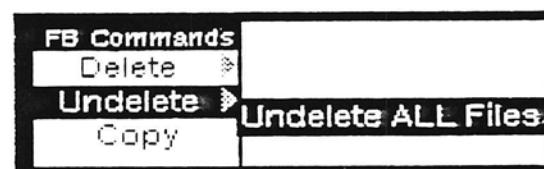


Figure 6. The Undelete All Files subcommand

The Delete command has a useful subcommand, Delete Old Versions (see figure 7). When you have been editing a file in the text editor and performing repeated Put commands, or you as a

programmer have done many MAKEFILEs of the same file, multiple versions of the file accumulate, each more recent version denoted by a higher number. The Delete Old Versions command is used to delete excess versions of the files displayed in the browser.



Figure 7. Delete

subcommands

To use this command, click the mouse down on the Delete command and roll the cursor out to the right, choosing the Delete Old Versions command. Unlike the Delete command (or Delete Selected Files, the equivalent subcommand), the Delete Old Versions command operates on *all* the files in the browser. The File Browser prompts you for the number of versions of each file that you wish to retain. It offers the default of one version. You can accept the default or you can type a different number of your choosing, followed by a carriage return. The File Browser then marks for deletion all but the most recent *N* versions of all

2

Compiling Files

1

, where *N* is the number you specified. Before issuing the Expunge command, you can, if you wish, scroll through the browser undeleting any particular files for which you wish to retain more versions than you specified.

The Delete Old Versions command is sometimes useful even when you are not planning to actually expunge the files. This is because of the way extending the selection avoids deleted files (see "Selecting Files"). For example, if you wanted to copy only the most recent version of all the files in the browser to another location, you could do the following: Use the Delete Old Versions command, retaining just one version. This marks deleted all files but the newest version of each. Now go to the start of the browser and select the first file, then scroll to the end of the browser and click the right mouse button to extend the selection to the end of the browser. You have selected exactly the newest version of each file. You can now use the Copy command to copy those files. Finally, use the Undelete All Files command to undelete all the old versions.

Expunging Files

If you are sure you want to delete a file or files permanently, choose the Expunge command. The Expunge command is

grayed while the File Browser expunges the files. As each file is removed from the system, it is removed as well from the browser display, and the tally of total number of files and number of deleted files is updated, so you can see the progress of the command.

If for some reason a file could not be expunged, the File Browser prints a message saying so in its prompt window, but continues to expunge the other files. The main reasons that prevent a file from being expunged are its being opened, either by you or some other user, or your not having the access rights required to delete it (if it is on a file server).

Note that the Expunge command is not affected by the current selection; it operates only on files marked for deletion, whether selected or not.

Copying Files

The Copy command is used to copy an entire file or set of files to another file system location, for example, from your disk to a file server. Select the file(s) you wish to copy, then click on the Copy command. The File Browser prompts you to supply a destination.

If you selected just one file, the File Browser prints the old name and offers as a default the same file name, but on the connected directory (see figure 8). You can accept the default or supply your own destination file name. If you supply just a directory specification, e.g., {SERVER}<YOURDIRECTOR>, the file is copied to that directory under its current name. If you supply a complete name, the file is copied to that exact name.

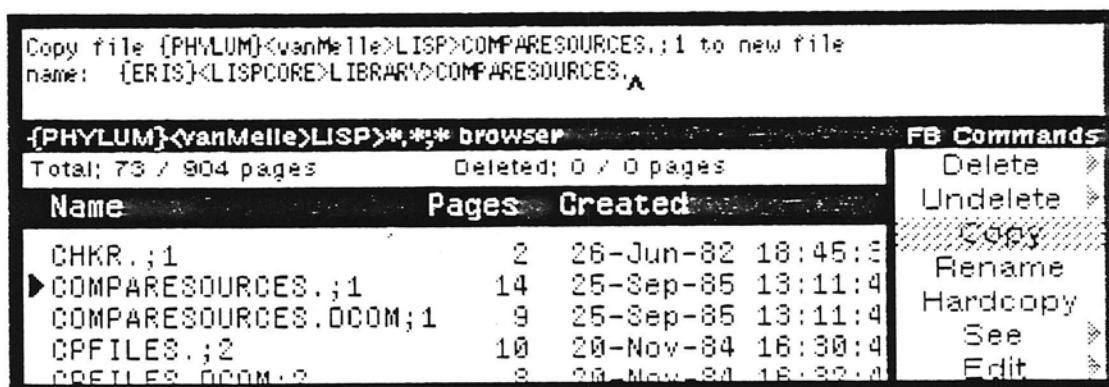


Figure 8. A File Browser showing input for copying a single file

If you selected several files, the File Browser notes how many files you wish to copy and offers as a default destination the connected directory. You can accept the default or supply a different directory. All the files are copied to that directory under the names they currently have. Note that you must supply a directory specification, e.g., {SERVER}<YOURDIRECTOR>,

rather than a complete file name, since you can't copy multiple files to the same name. If you mistakenly type a file name, rather than a directory specification, the File Browser will complain and abort the command.

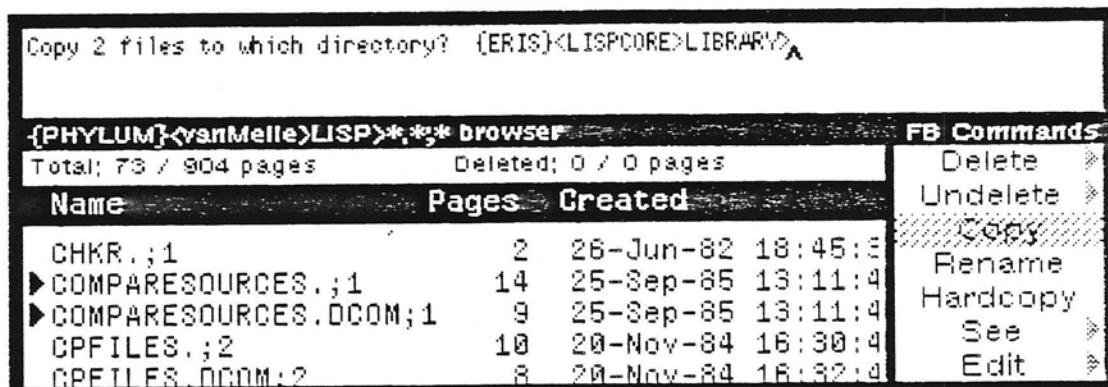


Figure 9. A File Browser showing the prompt for copying a group of files

As each file is copied, the File Browser prints a message giving the full name of the new file. If a file with the chosen name already exists, the new file's version number will be one higher; otherwise it will be version one. T

new file will have the same creation date as the original file. If the destination file happens to be one that matches the pattern of the files in the browser, the new file is inserted in the appropriate place in the browser display. However, if it matches the pattern of some other File Browser, it is not inserted in that other browser's display (in other words, File Browsers do not, in the current implementation, know about each other).

Moving and Renaming Files

The Rename command is used for changing the name of a file or group of files. If the new name of the file is on a different file system than the original file, changing its name is equivalent to moving the file, i.e., copying it to the new name and then deleting the original file.

The Rename command is used in exactly the same way as the Copy command. If you rename a single file, you can supply a complete new name or just a directory; if you rename several files, you must specify a directory. As each file is renamed, the File Browser prints a message giving the file's new name and removes the file from the browser display. If the new name belongs in the same browser, it is inserted in the appropriate place. If for some reason a file could not be renamed, this is noted in the prompt window. The reasons for the failure of a renaming operation are roughly the same as for the failure of an Expunge—the file is open, or you do not have the access rights needed to rename the file.

Loading Files

The File Browser's Load command can be used to load both source (interpreted) and compiled files into your workstation's virtual memory. First select the file or files you want to load, then select Load with the left mouse button. A special display window is opened to give information about the files as they are loaded (see figure 10). When the load is complete, the File Browser closes the Load window.

```
TTY window for LOAD
{ERI8}<LISPUSERS>CROCK.DCOM;2
compiled on 21-Jun-86 13:26:30
FILE CREATED 21-Jun-86 13:25:30
CROCKDCOMS
```

Figure 10. The Load window

The Load command also has subcommands that enable you to load files in different ways. These commands are described briefly here; see the *Interlisp-D Reference Manual* for further details. Only the Load and Load Sysload commands are of interest to nonprogrammers. All load commands are placed on the history list. With the exception of Load Sysload, all are undoable using history list commands.

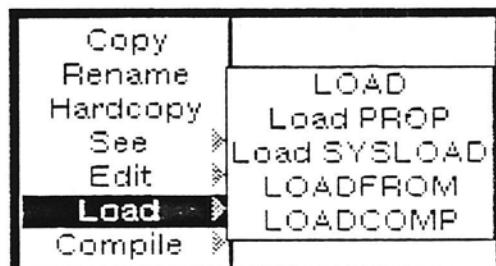


Figure 11. Various ways to load a file

Load (same as the main Load command) loads the file(s) with LDFLG = NIL. If any functions or variables in the file redefine ones that are already in memory, messages such as "FOO redefined" are printed. The command is undoable. This command is used for loading source files that you as a programmer plan to make modifications to, or for loading any file that you think you have doubts about, in which case you might want to be able to undo the load.

Load Prop loads the file(s) with LDFLG = PROP. Functions are loaded onto property lists, rather than redefining the functions already in memory. This command is used for loading Interlisp-D source files for which the compiled version already exists in memory, but which you plan to edit.

Load Sysload loads the file(s) with LDFLG = SYSLOAD. Function and variable redefinitions occur quietly. The file package is not informed of this file. This is the fastest loading command and consumes the fewest resources, but it is not undoable. It is the best way to load compiled (extension DCOM) files that you are certain you want to load into your environment and are not planning to edit.

Loadfrom calls the file package's function LOADFROM. This loads variables and other expressions but not functions, and does so in a way that informs the file package, so that the editor knows where to find the functions.

Loadcomp calls the file package's function LOADCOMP. This loads compile-time expressions from the file(s).

Compiling Files

The Compile command is used to compile a selected Lisp source file or files. The files do not have to be loaded. The Compile command uses BCOMPL, the block compiler, to compile the files, unless you select TCOMPL from the Compile submenu. This command is placed on the history list, so that it is undoable.

A special executive window is opened for each file to display information about the code being compiled. When the compilation is finished, the window is closed. Each compiled file is saved on your connected directory with the original file name and the extension DCOM.

Note that this command compiles a file or files found on a storage device, not the functions defined in the Lisp image. If you have made changes to any of the functions on a loaded file, you must perform a MAKEFILE to write an updated version before compiling it. For more information on making files and compiling, see the *Interlisp-D Reference Manual*.

usually eight-bit) bytes.

Page

The number of 512-byte pages in the file.

ByteSize.

The size (in bits) specified pattern at the time you created the browser. Although the File Browser is aware of files copied,

renamed, or expunged within the browser, it is completely unaware of any changes to the file system that happen outside the browser. If you have reason to believe the file system has changed, and you want the browser to reflect the latest state of the file system, use the Recompute command. For example, if you open a File Browser on your directory, then save several versions of a TEdit file on that directory, the file listing will not display the new versions.

The Recompute command operates exactly as when you started up the File Browser initially: it clears the display and tally windows, then enumerates the files matching the pattern. The Recompute command in the menu is grayed until the enumeration is finished. During this time you cannot scroll or perform any other operations on the browser. However, you can close the window if you want to abort the command and throw away the browser.

The Recompute command also has a menu of subcommands that allow you to list different files or different information for the same set of files (see figure 12). To obtain this submenu, hold the left mouse button down on Recompute and roll the mouse cursor out to the right.

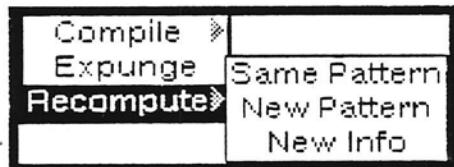


Figure 12. Recompute subcommands

Same Pattern is the same as the main Recompute command, i.e., it enumerates the files matching the same pattern as before.

New Pattern lets you change the pattern, i.e., browse a new set of files. The File Browser prompts you to supply a new file name pattern, and offers the old pattern as an initial default (see figure 13). You can either type an entirely new pattern, replacing the one offered, or delete the old pattern, one character at a time, by backspacing. Press the carriage return when you have finished specifying the pattern. The File Browser then enumerates the files matching this pattern, just as with the Recompute command. You can abort the pattern specification, and thus the whole command, by erasing the whole pattern (by backspacing or using control-Q) and then pressing the carriage return.

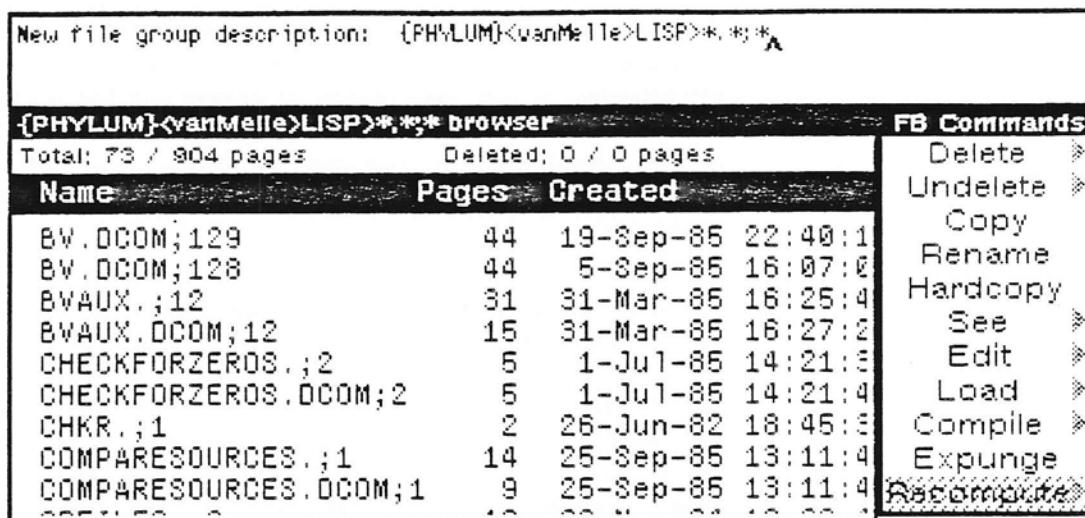


Figure 13. The

prompt for a new file name pattern

New Info lets you change which types of information you want displayed for each file. It is described in the next section.

Getting Information About Files

The File Browser displays some file *attributes*, or information you can give the Recompute command.

When you try to expunge a file and the File Browser displays the message "Can't expunge FILENAME," the file is probably still open. You can get a list of open files by typing (OPENP) at the prompt in your rs by changing FB.DEFAULT.INFO (see "Customizing the File Browser"). You can change the attributes displayed in a particular browser window by using the New Info command.

To use the New Info command, select it from the roll-out submenu of the Recompute command. The File Browser opens up an additional subwindow, the Info Options window, below the display window. This subwindow contains a menu of attributes, with the current defaults shaded (see figure 14). Clicking on a shaded item unshades it; clicking on an unshaded item shades it. When you have selected all the attributes you wish to see displayed, issue either the Recompute or the New Pattern command. The files will be listed with the new information you requested. The Info Options window stays open—you can close it at any time.

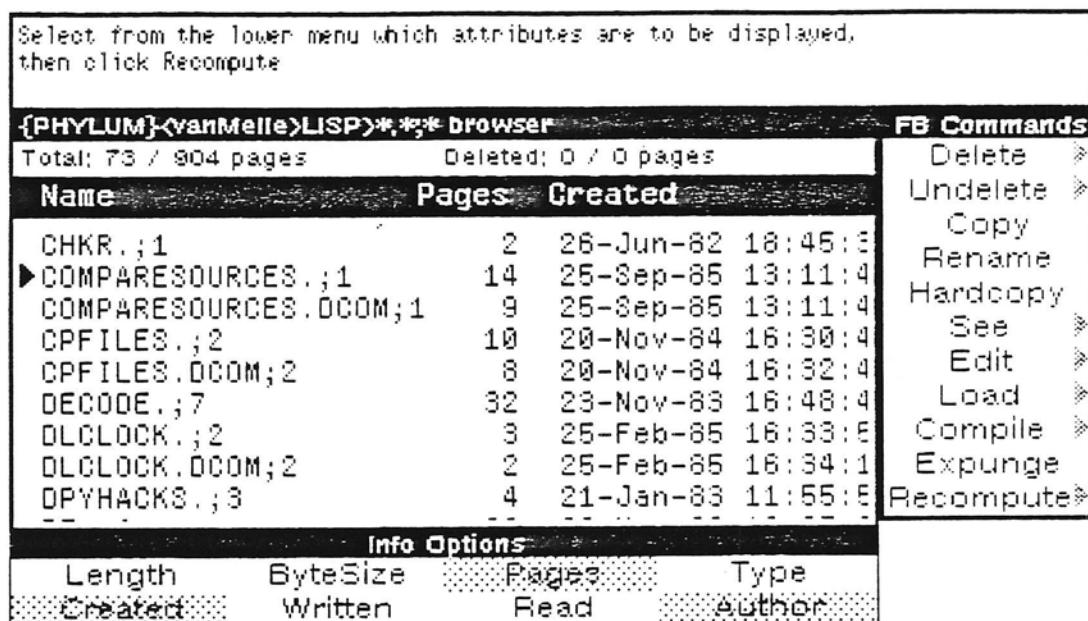


Figure 14. A File Browser showing the Info Options subwindow

The Info items have the following meanings:

Created.

The date and time that the content of the file was created. This date changes whenever the file is modified, but does not change when a file is copied or renamed.

Written.

The date and time the file was last written to the file system. This date is never older than the Created date, but it can be newer if the file is copied, unmodified, from one file system to another.

Read.

The date and time the file was last read. This attribute is usually blank if the file has never been read.

Author.

The log-in name of the person who wrote the file, or last modified it.

Length.

The length of the file in (usually eight-bit) bytes.

Pages.

The number of 512-byte pages in the file.

ByteSize.

The objects returned in the file. In Interlisp-D this is always eight, but some older computers and file servers allow other sizes.

Type.

A value indicating what kind of data the file contains. The usual values of this attribute are TEXT, meaning the file contains just characters, or BINARY, meaning the file contains arbitrary data. Some servers have additional types, such as INTERPRESS for files in Interpress format.

Quitting the File Browser

To quit a File Browser, simply close the browser window. If any files have been deleted but not expunged, a small menu will pop up listing two options: Expunge Deleted Files and Don't Expunge (see figure 15). If you choose the Expunge Deleted Files command, the files will be expunged before the window closes. If you choose the Don't Expunge command, your deletions are ignored. If you click outside the menu, no action is taken and the Close command is aborted.

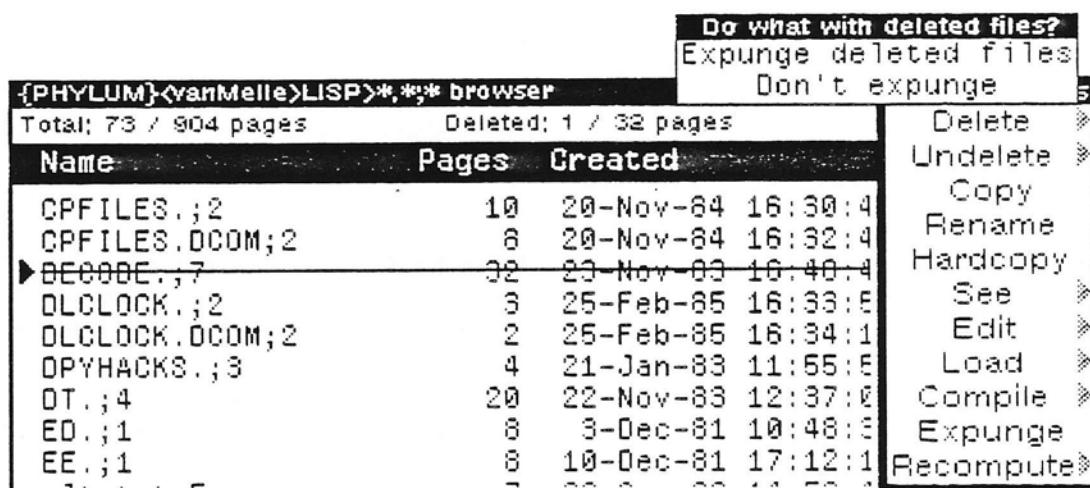


Figure 15. The menu of Expunge options

If you are finished with the File Browser only temporarily, and want to put it aside to work on later, you can shrink the browser. The browser shrinks to an icon displaying the pattern of files inside the browser (see figure 16). If any files are marked for deletion, you will be prompted with the same menu of Expunge options as when you close a browser.

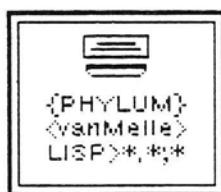


Figure 16. The File Browser icon

Troubleshooting Problems With the File Browser

When the File Browser returns the message "No files in group *FILENAMEPATTERN*" when you know those files exist, the file server is probably down or rejecting connections. If this is so, your only option is to wait until the server is functioning again, and then give the Recompute command.

When you try to expunge a file and the File Browser displays the message "Can't expunge *FILENAME*," the file is probably still open. You can get a list of open files by typing (OPENP) at the prompt in your executive window, and close an open file by typing (CLOSEF *FILENAME*). You can close all open files by typing (CLOSEALL).

Customizing the File Browser and Using the Programmer Interface

File Browsers are created by one of the following two functions:

(FILEBROWSER *FILESPEC DESIREDPROMS REGION*) [Function]

Creates a File Browser on files matching the pattern *FILESPEC* and displaying the attributes *DESIREDPROMS* for each file. If *DESIREDPROMS* is missing or NIL, it defaults to the value of FB.DEFAULT.INFO (below). If *REGION* is specified, it is a screen region in which to open the browser; otherwise, the user is prompted for a region. This function returns the main window created.

(FB *FILESPEC ATTR1 ... ATTRN*) [Nilambda Nospread Function]

This is the function provided for creating File Browsers by typing commands to the executive. FB creates a File Browser on files matching the pattern *FILESPEC* and displaying the attributes *ATTR1* through *ATTRN*, or the value of FB.DEFAULT.INFO if no attributes are specified. The user is prompted for a window region. For example, the executive command FB *.DCOM LENGTH CREATIONDATE browses all files on the connected directory with extension DCOM, displaying the length in bytes and creation date for each. FB always returns NIL.

There are several global variables that can be altered to affect the File Browser's behavior. You can set them by typing (SETQ *VARIABLENAME NEWVALUE*) to your executive window, and can save their values with a VARS command in your initialization file.

FB.DEFAULT.INFO [Variable]

A list specifying which attributes should be displayed for each file. The elements of this list are the Lisp names for the attributes you want displayed. The choices are CREATIONDATE, WRITEDATE, READDATE, LENGTH, SIZE (corresponds to the Info item Pages), BYTESIZE, AUTHOR, and TYPE. For example, (SETQ FB.DEFAULT.INFO '(CREATIONDATE LENGTH)) would cause all

new File Browsers to display exactly the attributes creation date and length in pages for each file.

FB.DEFAULT.NAME.WIDTH [Variable]

The amount of space, in points, to use for displaying file names in a browser, initially 140. The name column is automatically expanded if enough names are too wide. You can set this larger if you routinely browse directories of long file names.

FB.ICONFONT [Variable]

The font in which the file pattern is displayed on the browser icon, initially eight-point Helvetica. The value of this variable should be a font descriptor, as returned by FONTCREATE. For example, (SETQ FB.ICONFONT (FONTCREATE 'MODERN 10 'BOLD)).

FB.BROWSERFONT [Variable]

The font in which the information in the main display window is printed, initially 10-point Gacha.

FB.PROMPTFONT [Variable]

The font in which prompt messages are printed, initially eight-point Gacha.

FB.PROMPTLINES [Variable]

The number of lines in the prompt window, initially three.

FB.MENU.ITEMS [Variable]

A list of the items that appear on the File Browser's command menu. You can add new File Browser commands by adding new elements to the end of this list. Element of FB.MENU.ITEMS is of the form (LABEL YOURFN "EXPLANATION"), where LABEL is the name of the command, as it is LABEL YOURFN "EXPLANATION", where LABEL is the name of the command, as it is to appear in the menu; YOURFN is the name of the function to be called when the command is selected, and EXPLANATION is the explanation to be printed when the mouse cursor is held over the command. While YOURFN is executing, the menu command is grayed out, and the File Browser is "locked" so that no other commands or processes can access it. You can have subcommands as well if you make the menu command be of the form (LABEL YOURFN "EXPLANATION" (SUBITEMS item1 ... itemN)), where each itemJ is recursively of the same form as a menu command.

The File Browser calls YOURFN with four arguments: (YOURFN BROWSER KEY ITEM MENU), as follows:

BROWSER is the File Browser object in control of this browser window.

KEY is the mouse key pressed (Left or Middle).

ITEM is the menu item that was selected.

MENU is the command menu.

YOURFN can also be a list of two elements, (*FN ARG*), where *ARG* is an arbitrary value that is passed, unevaluated, as the fifth argument to *FN*. This is useful for writing one function that implements several subcommands that are similar to the original command.

Any additions to FB.MENU.ITEMS should be saved with the File Package command APPENDVARS, so that the new items are added to the end of the menu, and your changes will not interfere with any changes to built-in File Browser commands in new Filef the objects returThe following functions are useful within your command function:

(FB.SELECTEDFILES *BROWSER NOERRORFLG*) [Function]

Returns a list of file objects representing the files currently selected in *BROWSER*. If there are no selected files, this prints "No files are selected" in the prompt window, unless *NOERRORFLG* is true, in which case this function quietly returns NIL.

(FB.FETCHFILENAME *FILEOBJECT*) [Function]

Returns the full name of the file denoted by *FILEOBJECT*, one of the objects returned by FB.SELECTEDFILES.

(FB.PROMPTWPRINT *BROWSER X1..XN*) [NoSpread function]

Prints the strings *X1* through *XN* in *BROWSER*'s prompt window. The item T is printed as a carriage return (i.e., a command to go to a new line).

(FB.PROMTFORINPUT *PROMPT DEFAULT BROWSER*) [Function]

Prompt's for user input in *BROWSER*'s prompt window. *PROMPT* is the prompt string, *DEFAULT* is the default answer. Returns the user's input as a string, or NIL if there is no input, or it was aborted with control-E