Changes to Lafite in the version of May, 1989.

New files. There are 3 new files: LAFITEFOLDERS, LAFITESORT, MAILSCAVENGE. In addition, Lafite requires the file DATEPATCH for improved date parsing. These are all loaded automatically by LAFITE.

Folder hierarchy. [This feature derives from a package written by Mike Dixon.] You can now organize your folders into a hierarchy, so that the folder menus presented by **Browse** and **MoveTo** are easier to use. Folders are organized into *groups*. A folder can be in any number of groups. Groups can have subgroups. The top level of the folder menu is composed of all the groups that are not a subgroup of some other group, plus all folders that are not in any group. The **Edit Folder Hierarchy** command on middle-button Browse lets you define new groups and change existing ones.

Note: the folder hierarchy is saved on your Lafite.info file, along with folder names and form names. Older versions of Lafite do not know about this structure, but those of the last year will at least preserve it. However, if you delete folders and then come back to a new version of Lafite, the deleted folders may still appear in the hierarchy; you must manually delete them either by editing the variable LAFITE.FOLDER.STRUCTURE or by editing the file Lafite.info.

Sorting. There is a new command on the middle-button browser titlebar menu: **Sort by Date**. This command sorts the messages in the browser in order of their "Date:" fields. Messages lacking a Date field or having an unparseable Date field are sorted so as to remain next to the preceding message. There is a subcommand **Sort Selected Range**, which only sorts between the first and last selected messages. In either case, the messages are sorted in the browser only, until you next **Update**, at which time the messages are written to the file in the sorted order. When messages have been rearranged, the **Update** option **Write Out Changes Only** is not available, as there is no file format for out of order messages; you must **Expunge**, even if there are no deleted messages.

Sorting New Mail. You can have Lafite automatically sort new mail by setting the variable LAFITE.SORT.NEW.MAIL. If the value is T, Lafite always sorts new mail; if :MULTIPLE, Lafite only sorts when the mail was retrieved from more than one server. Note that even when using a single mail server, mail can easily be out of order, due to the varying speeds at which messages pass through gateways, etc. Sorting assures that the messages will appear in the order in which they were posted. Sorting new mail seems to be fast enough (at least on my Dorado) that it is probably always worth doing.

Dates. As you might guess from the above, Lafite now actually parses the Date field of messages. Aside from enabling Sort, this allows Lafite to display the date in the browser in a canonical form, rather than one that varies with the whim of the sender's software. Lisp's date parser (IDATE) was substantially beefed up to handle a wider variety of dates. It now claims to handle all dates legal in RFC822 syntax (except the silly single-digit military time zones), plus a fair variety of other formats found in mailers of recent years. The parser changes are in a module DATEPATCH loadable separately from Lafite.

Lafite considers dates older than 1970 spurious, so as to avoid being fooled by messages from machines that neglected to set the time.

With a new representation of dates, Lafite also has a new table of contents format. If you browse a folder with an old-style toc, Lafite will print "(older format)" when it reads it. Next time you do an Update, the toc will be written in the new format. Lafite does not automatically parse dates on old messages, so until you issue a Sort command, old message dates will still be displayed as whatever string the old toc saved, not the new canonical form. In addition, when you *do* issue the Sort command, it must first go back and retrieve afresh all the date fields and parse them. Old versions of Lafite cannot read the toc files saved by this new version of Lafite, so if you try to browse a new folder with an old Lafite it will be forced to parse the file from scratch.

Scavenger. The mail scavenger has been completely rewritten. It now handles most simple cases of bad message lengths by altering the file in place, rather than laboriously copying the entire file to a scratch location. The diagnostic output is, I hope, more informative. The scavenger is also integrated with the

browser, so that if you browse a malformed folder and the browser reports "unable to parse", you are immediately offered the option of scavenging.

Expunge. Lafite now gets less confused if you abort it in the middle of an Expunge. Either you killed it before it got to the point of irrevocably altering the file, in which case it is as if you had never started the Expunge, or it will report that the file is inconsistent and must be rebrowsed.

Long messages. The Lafite message file format has been very slightly extended to permit you to receive messages of up to 99,999,999 bytes (about 100MB!) in length. This means it no longer is forced to split apart messages longer than 99,999 bytes, which did very bad things to large Viewpoint attachments and sometimes caused loss of mail server connection. It also means that Lafite files are no longer exactly compatible with Laurel or Hardy format, in case anyone cares. As mitigation, however, messages that fit in the old format (99,999 bytes or less) are converted back to the old format whenever they are moved (either by **MoveTo** or **Expunge**), so you can arrange for a Hardy-compatible file by moving all messages into another folder (assuming your messages are all short enough).

NS Mail. Lafite accepts messages sent in serialized format 3, which it turns out is identical to format 2.

Lafite now sends plain text messages as "text attachments" rather than "mail notes", unless *NSMAIL-SEND-MAIL-NOTES* is true. This avoids a nasty bug in Services 11.3 mail servers.

An incomplete and not very well tested feature: you can send attachments. The attachment must be a file on an NS file server. To send an attachment, place a line "Attached-File: *XNS Filename*" in the header of your message. To send the file as a reference (this sends only a pointer to the file, not the file's content, so the recipient must have access to your server), use the line "Attached-Reference: *XNS Filename*". You can have only one such line in the header, and the body of the message must be small enough to send as a mail note (less than 8000 characters). Probably the only interesting kind of attachment to send currently is an Interpress master, which both Lafite and Viewpoint recipients will be able to print. To "forward" an attachment you received in NS mail, you can use **Put to File** on the attachment, then compose a message with "Attached-Reference" or "Attached-File" referring to the resulting file. Incompleteness: Most of the attributes of the file are lost when you send the file itself as an attachment, rather than a reference. You cannot send directories as attachments.