UNIXMAIL

By: Bob Bane (Bane.mv@envos.Xerox.com)

INTRODUCTION

UNIXMAIL is a new mail sending and receiving mode for Lafite. It sends mail via Unix hosts using the SMTP mail transfer protocol and can receive mail either by reading a Unix mail spool file or by calling the Berkeley mail program.

INSTALLATION

Turn Lafite off, load the file UNIXMAIL, make sure UNIXMAIL is configured appropriately (check the settings of the variables below, and make sure any other modules UNIXMAIL may need are loaded), then restart Lafite. If you are running Lafite on a machine that is isolated from the Xerox mail environment, you will probably want to set the variable LAFITE.USE.ALL.MODES to NIL and call (LAFITEMODE 'UNIX) before you turn Lafite back on.

CONFIGURING

See **SENDING MAIL** and **RECEIVING MAIL** below for the exact meanings of the variables you will be asked to set.

D-machines:

UNIXMAIL.SEND.MODE must be set to SOCKET and UNIXMAIL.SEND.HOST must be set to the name of a TCP host that will accept SMTP connections. UNIXMAIL.RECEIVE.MODE must be set to SPOOL and UNIXMAIL.SPOOL.FILE must be set to the pathname of your Unix mail spool file.

Unix-based emulators:

The default values of UNIXMAIL.SEND.MODE and UNIXMAIL.RECEIVE.MODE (PROCESS and SPOOL, respectively) will work if you normally send and receive mail from the machine where Medley is running.

OTHER MODULES YOU MAY NEED

UNIXMAIL may need other library modules to work. The modules needed vary depending on what hardware you are using:

D-machines:

TCP is mandatory for Unix sending and may be used for Unix receiving, NFS is optional for Unix receiving

Unix-based emulators:

one of TCPOPS or UNIXCOMM is mandatory for sending

SENDING MAIL

UNIXMAIL can send mail in one of two ways, depending on the setting of UNIXMAIL.SEND.MODE:

UNIXMAIL.SEND.MODE [Variable]

If its value is the atom PROCESS, UNIXMAIL will send mail by doing SMTP with a Unix process-stream, normally running `/usr/etc/mconnect`. This option only works on Medley running one of the Unix-based emulators.

If its value is the atom SOCKET, UNIXMAIL will send mail by doing SMTP with a TCP host. For this to work, an appropriate version of TCP must be loaded: either the TCP library module for D-machines or the TCPOPS library module for emulators that support it.

UNIXMAIL.SEND.MODE defaults to PROCESS.

Each of these send modes can be configured as well:

UNIXMAIL.SEND.PROCESS [Variable]

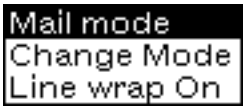
When UNIXMAIL.SEND.MODE is PROCESS, the value of this variable is the program run to create the SMTP process-stream. Initially the string `"/usr/etc/mconnect"`

UNIXMAIL.SEND.HOST [Variable]

When UNIXMAIL.SEND.MODE is SOCKET, the value of this variable is the name of the host UNIXMAIL will attempt to contact via TCP to open an SMTP stream over socket 25. Initially NIL; on a Unix-based emulator this means to try the machine Medley is running on. This variable must be set when running on a D-machine.

UNIXMAIL.WRAP.LINES [Variable]

This flag controls whether or not outgoing mail has its lines word-wrapped to a fixed length. It defaults to T, meaning word-wrapping is done. UNIXMAIL patches the Change Mode menu entry of the standard Lafite message form, adding an entry for toggling UNIXMAIL.WRAP.LINES:



UNIXMAIL.WRAP-LIMIT [Variable]

If UNIXMAIL.WRAP.LINES is true, this variable is the length in characters to which lines are wrapped. Default value is 72.

UNIXMAIL.TABWIDTH [Variable]

If UNIXMAIL.WRAP.LINES is true, this variable is the width tab characters are assumed to expand into for word-wrapping purposes. Default value is 8.

UNIXMAIL.RECIPIENT.PATTERNS [Variable]

This variable is a list of patterns that are applied to outgoing UNIXMAIL addresses; it can be used to catch bogus addresses and modify them before sending. List entries are of the form (*pattern* . *function*) where *pattern* is a list of arguments that will be passed along with the address to STRPOS; if STRPOS returns non-NIL, *function* is called with the address and the result replaces the address. For example, if mail from UUCP host `black-silicon` is arriving via path `mimsy!black-silicon`, but the address in its headers is missing the `mimsy`, this entry on UNIXMAIL.RECIPIENT.PATTERNS will add it back:

```
(("black-silicon!" NIL NIL T) LAMBDA (R) (CONCAT "mimsy!" R)
```

This means that whenever `(STRPOS "black-silicon!"` *`address`* `NIL NIL T)` returns non-NIL, *`address`* will have `"mimsy!"` prepended before the message is sent.

RECEIVING MAIL

UNIXMAIL can receive mail in one of two ways, depending on the setting of UNIXMAIL.RECEIVE.MODE:

UNIXMAIL.RECEIVE.MODE [Variable]

If its value is the atom SPOOL, UNIXMAIL will receive mail by reading a Unix mail spool file.

If its value is the atom MAILER, UNIXMAIL will receive mail by running a Berkeley mailer as a Unix process-stream, normally `/usr/ucb/mail`. This option only works on Medley running one of the

Unix-based emulators, and is a bit slower than SPOOL mode; it is primarily useful when you wish to occasionally switch between Lafite and the Berkeley mailer.

UNIXMAIL.RECEIVE.MODE defaults to SPOOL.

Each of these receive modes can be configured as well:

UNIXMAIL.RECEIVE.PROCESS [Variable]

When UNIXMAIL.RECEIVE.MODE is MAILER, the value of this variable is the program run to create the SMTP process-stream. Initially the string `"/usr/ucb/mail -N"`; the `-N` means to not print any banner or read any intialization file on starting the mailer.

UNIXMAIL.DONT.RECEIVE.STATUS [Variable]

When UNIXMAIL.RECEIVE.MODE is MAILER, the value of this variable is a set of message status letters; UNIXMAIL will leave behind any message whose status is included**.** Initially `""`, which means to read all messages regardless of status; another useful value would be `"O"` which means leave old messages behind.

UNIXMAIL.SPOOL.FILE [Variable]

When UNIXMAIL.RECEIVE.MODE is SPOOL, the value of this variable is the file UNIXMAIL will receive mail from. Any time this file has characters in it, Lafite will say you have new mail; when Lafite gets mail from this file, it will read all messages in the file and then set its size to zero. Initially NIL; on a Unix-based emulator this means to try the file `"{UNIX}/usr/spool/mail/`**`username`**`"`, where **`username`** is the value of (UNIX-USERNAME). To access a Unix mail spool file from a D-machine, it will probably be necessary to load and configure either the TCP or NFS modules and then set UNIXMAIL.SPOOL.FILE appropriately.