NAME

mail_open, mail_priority, mail_abort, mail_close, mail_close_alternate, mail_alloc, mail_free, mail_host − zmailer message submission interface

SYNOPSIS

#include <stdio.h>
#include <zmailer.h>

FILE *mail_open(char *)

int mail_priority;

int mail_abort(FILE *)

int mail_close(FILE *)

int mail_close_alternate(FILE *mfp, char *where, char *suffix)

char *mail_alloc(unsigned int)

int mail_free(char *)

char *mail_host()

The main program against which this library is liked is expected to export
  char *progname;
variable which carries a copy of *argv[0]* or something else that programmer wants to set.

Linkage time: **-lzmailer**

DESCRIPTION

These routines may be used by any program wishing to submit mail. *mail_open* will return a **FILE *** to a message file that should be written to by the application. This message file contains three parts: the message envelope, the message header, and the message body. The exact format of these components depend on the message protocol, which must be specified as the parameter to *mail_open*. The choices are predetermined by the capabilities of the mailer, and are defined in the header file. The known possibilities are:

MSG_RFC822
: this is the only format supported by default by the mailer. The message headers and body in this format are defined by the DARPA Request For Comments 822 and 1123. The message envelope syntax is similar to the message header syntax, but for complete details refer to the ZMailer documentation.

MSG_FAX
: intended for fax transmissions.

MSG_UUCP
: intended for old style UUCP format message headers

MSG_X400
: intended for X.400(88) messages.

The *mail_open*() routine will look for **FULLNAME** and **PRETTYLOGIN** environment variables and translate them into message envelope data for use by the mailer if it generates a sender address header for the message.

Note that the return value from the *mail_open*() routine corresponds to the return value of an *fopen*(3), and similarly the return values from *mail_abort*() and *mail_close*() correspond to the return value of *fclose*(3).

The *mail_priority* variable has default value of 0, and is used on scanning Zmailer configuration variable **ROUTERDIRS**, which tells alternate router directories under the **POSTOFFICE** directory. At value 0, **ROUTERDIRS** variable is not used. At higher values, successive directory from "ROUTERDIRS" is taken. See below about Z−Environment.

The *mail_close_alternate*() can be used to send currently open message file to some alternate destination, and is used at *smtpserver*(8zm) to send some quick-action requests directly to the *scheduler*(8zm).

The *mail_alloc*() and *mail_free*() routines are used to provide memory space for internal data structures. The versions of these routines in the library simply call *malloc*(3) and *free*(3) but an application may override them if desired.

Similarly the *mail_host*() routine is intended to return a unique string for each host, by default the hostname, and this too is intended to be overridden by an application that may already have this information available in some form.

ENVELOPE HEADER LINES

The message envelope headers are used to carry meta-information about the message. The goal is to carry transport-envelope information separate from message (RFC-822) headers, and body.
At first the message starts with a set of envelope headers (*-prefix denotes optional):

```
*external \n
*rcvdfrom %s@%s (%s) \n
*bodytype %s \n
*with %s \n
*identinfo %s \n
Either:
  from <%s> \n
Or:
  channel error \n
*envid %s \n
*notaryret %s \n
```

Then for each recipient pairs of:

```
*todsn [NOTIFY=...] [ORCPT=...] \n
to <%s> \n
```

Just before the data starts, a magic entry:

```
env-end \n
```

Then starts the message RFC-822 headers, and below it, the body

EXAMPLE

```
... set up signal handlers ...
FILE *mfp = mail_open(MSG_RFC822,0);
if (mfp != NULL) {
        ... output the mail message to mfp ...
} else
        ... error handling for not being able to open the file ...
if (some application processing went wrong
                || we took an interrupt)
        (void) mail_abort(mfp);
else if (mail_close(mfp) == EOF)
        ... error handling if something went wrong ...
```

ENVIRONMENT VARIABLES

FULLNAME
: variable defines textual fullname, for example: "Sample User"

PRETTYLOGIN
: variable defines **user@node** format of what user wants to claim as his/her own address (it must match those of mail router accepts.)

LOGNAME

USERNAME

USER
: Environment variables picked up in sequence in order to find an idea about user's identity to be stored into a line like:

  recvfrom STDIN (username@hostname)

  which is purely cosmetic documenting thing.

Z−ENVIRONMENT VARIABLES

POSTOFFICE
: defines directory where all POSTOFFICE functions are under.
  Example: **POSTOFFICE=/var/spool/postoffice**

ROUTERDIRS
: defines a ':' separated list of alternate router directories. If these are defined at all, they **must** exist, if alternate queueing priority mechanism is desired to be used.
  Example: **ROUTERDIRS=router1:router2:router3:router4**

ROUTERDIRHASH
: Value "1" at this variable means that all router directories **must** have subdirectories with names "A" thru "Z", alike

ROUTERNOTIFY
: defines an *AF_UNIX/DGRAM* type local notification socket into which a receiving client *may* inform the *router*(8zm) that there is some new job available.

INPUTDIRS
: *Presence* of this ZENV-variable means that messages will be tried to be stored into these alternate directories (syntax alike with ROUTERDIRS !) so that it can be leisurely analyzed (by e.g. virus scanners) before feeding it onwards to virus scanners.

  Presence of this also activates checking of *INPUTDIRHASH*, and *INPUTNOTIFY* .

INPUTDIRHASH=1
: This is analogous to *ROUTERDIRHASH* .

INPUTNOTIFY
: This is analogous to *ROUTERNOTIFY* .

FILES

/var/spool/postoffice/public/
: $POSTOFFICE/public/

/var/spool/postoffice/router/
: $POSTOFFICE/router/

/var/spool/postoffice/{ROUTERDIRS}/
: $POSTOFFICE/{ROUTERDIRS}/

/var/spool/postoffice/.router.notify
: $POSTOFFICE/.router.notify

SEE ALSO

sendmail(8zm), *zmailer*(1zm), *zmailer.conf*(5zm).
RFC-2822

AUTHOR

This program authored and copyright by:
Rayan Zachariassen <no address>
ROUTERDIRS mechanism is by:
Matti Aarnio <mea@nic.funet.fi>