The Wayback Machine - https://web.archive.org/web/20060909000649/http://www.eudora.co...

E U D O R A ™

SEARCH   CONTACT US

PRODUCTS | SUPPORT | USER FORUMS | AFFILIATES | DEVELOPERS | PRESS | TESTIMONIALS

QUALCOMM

Eudora Commandline Interface

Eudora for Windows supports a command line interface with which you can queue externally generated messages.

- **Queueing Externally Generated Messages from the Command Line**

- **Message Formats for Externally Generated Messages**

Queueing Messages with Eudora's Command Line Interface

Windows Eudora is capable of queuing and sending externally-generated messages as specified via the command line. If Eudora is started with a command line that specifies a fully-qualified path ending with an ".msg" extension, Eudora will treat the specified path as an externally-generated message and queue it for delivery.

For example:

    c:\eudora\eudora.exe c:\incoming\one.msg

will cause Eudora to process and queue one.msg to the Out mailbox.

If an existing instance of Eudora is already running, the new instance of Eudora will forward the request and terminate. If an existing instance of Eudora is not running, the new instance be process the request and then continue to execute.

A copy of the last "non-command-oriented" command line is cached in the registry. New instances of Eudora processing "command-oriented" command lines revert to the cached command line. (Other "commands" include specification of attachments, stationery, and mailto requests.) Hence, if Eudora is first started as follows:

    c:\eudora\eudora.exe c:\mail // use c:\mail as working directory

then restarted as follows:

    c:\eudora\eudora.exe c:\incoming\one.msg

Eudora will process and queue one.msg using c:\mail as it's working directory.

A similar result can be acheived on the Mac using Apple Script and Apple Events.

Back to the Top

Message Formats for Eudora Command Line Interface

Externally-generated messages may be either text-based or MIME encoded. During processing of the .msg file Eudora scans for headers. If a MIME-Version header is detected Eudora assumes the message has already been MIME encoded.

(Headers typically include To:, From:, Subject:, etc. Headers are separated from the message body by a single blank line.)

The mechanism allows any kind of MIME message to be sent, even ones that can not normally be generated by Eudora. It is the user's responsibility to make sure the message sent adhere's the the MIME standard.

MIME encoded messages are handled as follows:

1. All headers are extracted from the .msg file and saved in Eudora's out.mbx and out.toc files.

2. A message summary is created and added to the Out mailbox Table of Contents.

3. The body of the .msg file is saved to a temporary file. This temporary file is treated as a attachment until the message is sent. The X-Attachment header of the message identifies the temporary file. (The original message body is not modified in any way.)

4. The user may view and modify the message headers in similar fashion to any other queued message. Since the original body is treated as an attachment it will not be visible. Any text added to the body of the message will be ignored when the message is sent.

5. Upon sending, the message is reconstructed using the headers from the out.mbx file (subject to modification), and the original .msg body (not subject to modification).

Text-based messages are handled as follows:

1. The headers from the original .msg file are parsed and saved in Eudora's out.mbx and out.toc files.

2. The body of the original .msg file is also saved in the out.mbx file.

3. A message summary is added to the Out mailbox table of contents.

4. The user may view and modify the message in similar fashion to any other queued message. The message body is visitable and modifiable.

5. Upon sending the message is treated as a normal Eudora message. Both the message headers and body are retrieved from the out.mbx file.

Back to the Top

| Home | Products | Support | Affiliates | Developers | Forums | Contact Us |
| QUALCOMM | Investor Relations | Section 508 | Privacy Statement | Terms of Use |