

Appendixes A-D

Appendix A – Sources

Eudora Information

.i.Eudora Information;The information in this manual was correct at the time of printing. However, things happen very quickly in the electronic world, meaning that some of this information may already be out of date. For the very latest information about Eudora, send E-mail to eudora-info@qualcomm.com.

Update Announcements

.i.Announcements;If you would like to receive announcements about updates to Eudora, send E-mail to listserv@vmd.cso.uiuc.edu. The text of the message should consist of “sub eudora your-first-name your-last-name.” Be sure you don’t put anything else in the body of the message, including your signature.

Obtaining a POP Server

.i.POP:Server for;.i.POP:Server for:UNIX;If you would like to run a POP server on your own UNIX system, we suggest you use “.i.popper;popper.” Popper is available by anonymous ftp from <ftp.cc.berkeley.edu>.

.i.POP:Server for:VAX/VMS;VAX/VMS systems may try the “Multinet” package from TGV, or IUPOP3, available for anonymous ftp from <ftp.indiana.edu>.

.i.POP:Server for:VM/CMS;For VM/CMS users, there is a port of “popper” available for anonymous ftp from <vmd.cso.uiuc.edu> (cd to the “POPD” directory).

Ph Server Source Code

.i.Ph:Server source code;A server for the “Ph” protocol is available for anonymous ftp from <uxc.cso.uiuc.edu>. The filename is `pub/qi.tar.Z`.

Password Change Server

.i.Password:Changing;Server source code for;Three sample UNIX servers for Eudora’s “Change Password” command are available in the Eudora Q&A stack.

Network Products Installer

The current version of the Network Products Installer disk can be had by anonymous ftp to <ftp.apple.com>, `dts/mac/sys.soft/netcomm` subdirectory.

MacTCP

Contact Apple Software Licensing, sw.license@applelink.apple.com.

Spellswell

For information on obtaining Spellswell, contact Working Software, Inc. at:

P.O. Box 1844

Santa Cruz, CA 95061-1844

Phone: (408) 423-5696

Fax: (408) 423-5699

Appendix B – Shortcuts

Modifier Keys

Many operations in Eudora can be affected by holding down one or more “modifier” keys. Eudora uses the [shift], [option] and [command] keys as modifiers.

The [shift] key is usually used to “constrain” things – to keep them from happening or make them happen in a limited way.

-

[shift] + Opening Eudora stops Eudora from checking for mail at startup, even if you have a mail checking interval set.

-

[shift] + Reply instructs Eudora to copy only the selected body text to the new message. This is sometimes easier than having Eudora copy it all and then deleting what you don't want.

•

[shift] + To:, Cc:, or Bcc: in the Nicknames window does not bring the composition window to the front after inserting the name.

•

[shift] + Check Mail instructs Eudora to check for mail and resets the checking interval (normally, checking mail manually doesn't affect when the next automatic check is done).

•

[shift] + Print instructs the printer to print the selected text only.

The [option] key is usually used to make an operation to reverse option settings.

•

[option] + Insert a nickname inserts the full nickname expansion in the specified field.

•

[option] + finish a nickname inserts addresses instead of nickname.

•

[option] + Queue or Send button brings up Change Queuing dialog.

•

[option] + Save instructs Eudora that all open windows are to automatically reopen at the next startup.

•

[option] + Reply toggles the "Reply to all" option in the Settings dialog. If the Reply to all option is turned on, Option + Reply generates a reply to the sender only. If the Reply to all option is turned off, Option + Reply generates a reply to all addresses in the message header.

•

[option] + Transfer sends a copy of the current message to the selected mailbox and retains the original message in the original mailbox.

•

[option] + Open a message shows all headers.

The [command] key is also sometimes used as a modifier.

•

[command] + drop a document onto Eudora attaches the document to a message, even if the document is one Eudora could have opened.

Other Important keys

The arrow keys, depending on the settings in the Settings dialog, can move you from one message to another in a mailbox,

•

up/left arrow or [command] + up/left arrow takes you to the previous message.

•

down/right arrow or [command] + down/right arrow takes you to the next message.

The [enter] key performs several different functions:

•

It selects the outlined button in any dialog, alert, or window.

•

It opens the selected message(s).

•

It lets you begin or end editing the subject of a message.

The [esc] key (or [command] + .) stops any operation currently in progress.

If you have an extended keyboard, the following keys are also useful:

•

[help] turns Balloon Help on or off.

•

[F1] is Undo.

•

[F2] is Cut.

•

[F3] is Copy.

•

[F4] is Paste.

•

[del] deletes the character to the right of the insertion point.

•

[home] scrolls the window to the beginning.

-
- [end] scrolls the window to the end.
-
- [page up], [page down] scrolls up or down through the window.
- [command] + Key Equivalents
- [command] + [.]
Stops Eudora 's current action
- [command] + [A]
Select all
- [command] + [B]
Move window behind another window
- [command] + [C]
Copy
- [command] + [D]
Delete message
- [command] + [E]
Queue an outgoing message
- [command] + [F]
Open Find window (search for message)
- [command] + [G]
Find again
- [command] + [H]
Attach document to outgoing message
- [command] + [I]
Open In mailbox
- [command] + [K]
Make nickname
- [command] + [L]
Open Nicknames window
- [command] + [M]
Check mail
- [command] + [N]
New message
- [command] + [O]
Open message
- [command] + [P]
Print message
- [command] + [Q]
Quit Eudora
- [command] + [R]
Reply to message
- [command] + [S]
Save message without sending
- [command] + [T]
Send queued messages
- [command] + [U]
Open Ph window
- [command] + [V]
Paste
- [command] + [W]
Close message
- [command] + [X]
Cut
- [command] + [Z]
Undo

Appendix C – Mail Transport

Introduction

Eudora uses Simple Mail Transfer Protocol (SMTP) .i.Simple Mail Transfer Protocol;.i.SMTP;to transfer your outgoing mail to your SMTP server machine, which in turn uses SMTP to send your mail to the world at large. Mail from the world at large arrives on your Post Office Protocol (POP) server.i.POP server;, where it waits for Eudora to pick it up with Post Office Protocol, version 3 (POP3).i.POP3;. The mail Eudora sends and receives is constructed in accordance with RFC 822 and RFC 1341 (MIME).

Eudora mail transport overview

Outgoing Mail

When you send an E-mail message to someone, Eudora uses SMTP to send the mail to your local SMTP server computer. That computer then sends the mail to your addressee's computer, also (usually) by means of the SMTP protocol.

Why doesn't Eudora talk directly to your addressee's computer? For one thing, it would take a lot longer for your mail to leave your Macintosh, because your Macintosh would have to call up each addressee's computer and deliver your mail. For another, some computers are "hard to find;" it's much better to let another computer "hunt" for your addressee than to make your Macintosh do it. Finally, sometimes your addressee's computers won't be available when you want to send mail. The SMTP server handles this by holding your mail until the other computer is ready to accept it, eliminating the inconvenience of having unsent messages hanging around on your Macintosh.

Incoming Mail

When somebody sends you mail, other computers use the SMTP protocol to deliver the mail to your POP server. Your POP server puts mail in your "mail drop," where it stays until the Eudora program picks it up. When you check your mail, Eudora uses POP3 to pick up your mail and move it to your Macintosh.

Why doesn't Eudora use SMTP to receive your mail? .i.SMTP;.SMTP works best when the computers it knows about are always ready for mail. Unless you wanted to run Eudora and your Macintosh 24 hours per day, seven days a week, SMTP wouldn't work very well for you. It also doesn't work well in lab environments, where you might use any number of different Macintoshes.

More Information

If you want to know more about the Internet in general, consult the book *Internetworking with TCP/IP*, by Douglas Comer, 1988, Prentice-Hall ISBN 0-13-470154-2 025.

If you want to know more about SMTP, RFC 822, POP3 and MIME, the official standards are:

RFC 821, "Simple Mail Transfer Protocol," by Jonathan B. Postel

RFC 822, "Standard for the Format of Internet Text Messages," by Ned Freed and Nathaniel Borenstein

.i.RFC's;.RFC 1225, "Post Office Protocol, Version 3," by Marshall Rose

RFC 1341, "Multipurpose Internet Mail Extensions," by Dave Crocker

You can find the RFC's by anonymous ftp to nic.ddn.mil; see *Internetworking with TCP/IP* for details.

Appendix D – Dialup Eudora

Introduction

.i.Dialup;.Eudora is designed for use with MacTCP or for dialup to Cisco terminal servers. It can be used with other dialup connections as well. Complicated setups are likely to be unreliable, but if the setup commands to connect are simple, Eudora works well.

These instructions assume you are familiar with ResEdit, your communications equipment and your hosts. It is suggested that one person make these changes to Eudora and then redistribute the customized version to other users at your site.

Macintosh Requirements

To use Eudora over a dialup connection, you must have the .i.Communications Toolbox;.Communications Toolbox installed. System 7 has it built in.

Dialup Requirements

Eudora needs a "transparent" connection to your POP, SMTP, and (optionally) Ph servers. Transparent means primarily two things:

1.

Characters Eudora sends should NOT be echoed back to Eudora. Most systems do .i.Echo;.And dialups;.echo characters, so something special may need to be done to achieve this.

2.

You must pay close attention to how carriage returns are treated. UNIX systems routinely translate .i.Carriage return;.And dialups;.carriage returns into linefeeds; you must either disable this on your system or teach Eudora how to deal with it.

Navigation

.i.Navigation;.Eudora has a very rudimentary scripting system built in. This capability is called "navigation" to

separate it from the rather sophisticated connotations of “.i.Scripting:And dialups;scripting.”

Navigation is simple; Eudora uses a list of strings which it sends out the serial port one at a time. After each string, it waits until there is no output from the remote system for 2 seconds or until a given string is matched, after which Eudora continues with the next string.

There can be three sets of navigation strings contained in STR# resources: one is used when connecting (.i.Navigate In;Navigate In); one is used when disconnecting (.i.Navigate Out;Navigate Out); and one is used when switching from SMTP to POP (.i.Navigate Mid;Navigate Mid). You may use ResEdit to manipulate these resources. Any of the resources may be absent, in which case Eudora skips the navigation it would otherwise have done with that resource.

For each function, Eudora selects an appropriate resource in the following manner. First, it takes the name of the .i.Connection tool;connection tool currently in use (e.g., “Apple Modem Tool”); then, it appends a space and the current type of navigation (e.g., “Navigate In”). Eudora looks for an STR# resource with that name (e.g., “Apple Modem Tool Navigate In”). If that is not found, Eudora looks for an STR# resource named after the navigation type (e.g., “Navigate In”). Finally, it looks for specific resource ids (2400 for Navigate In, 2600 for Navigate Out, and 4200 for Navigate Mid). The first STR# resource found is the only one used.

Navigation resources can be in the Eudora application itself, in your Eudora Settings file, or in a .i.Plug-in;plug-in file. The latter is a file with type “.i.rsrc;rsrc” and creator “.i.CSOm;CSOm” placed in your Preferences folder; use of a plug-in file is highly recommended.

Plug-In files go in the Preferences folder.

Special Sequences in Navigation Resources

.i.Navigation:Special characters;Eudora provides a set of special character sequences for use in Navigation resources. These sequences are all two characters long, the first character being a backslash. They are either replaced with items from your Eudora settings, or they modify the Navigation process.

Replacements

These special characters are replaced with strings. They can appear at any place in a navigation string.

.i.\u;\u

.i.POP Account username;POP account user name.

.i.\h;\h

.i.POP Account hostname;POP account host name.

.i.\p;\p

.i.POP Account password;POP account password.

.i.\s;\s

.i.SMTP server hostname;SMTP server host name.

.i.\U;\U

.i.Dialin username;Dialin user name.

.i.\P;\P

.i.Dialin password;Dialin password.

.i.\n;\n

A .i.Linefeed;linefeed (ASCII 10).

.i.\r;\r

A .i.carriage return;carriage return (ASCII 13).

.i.\\;\

A single .i.backslash;backslash character.

Modifiers

.i.\b;\b

Hide from Progress window.

.i.\D;\D

.i.Delay;Delay.

.i.\B;\B

.i.Break;Break.

.i.\e;\e

.i.Expect;Expect something.

.i.\b;\b – when this appears as the first character in a given a string, Eudora won't print the string in the Progress window when it is sent. This is useful for passwords or other state secrets.

.i.\D;\D – should be the first character in the string and followed by digits. The digits are taken as a number of seconds to .i.Delay;delay (e.g., “\D2” is a two-second delay). As with other strings, Eudora will wait for output to stop for two seconds, or for an expect string, before proceeding.

.i.\B;\B – should be the first character in the string and followed by digits. The digits are taken as the number of ticks (60ths of a second) to send a .i.Break;break signal (e.g., “\B30” is a half-second break). As with other strings, Eudora waits for output to stop for two seconds, or for an expect string, before proceeding.

.i.\e;\e – makes Eudora .i.Expect;expect to see a specific string in the output from the dialup server. Eudora sends

whatever comes before the \e, and then waits for whatever comes after it to occur in the data sent from the host. There are two caveats to this. First, Eudora only matches on the first 7 bits; the high bit of each character is ignored for matching. This is needed for systems that use parity. Second, Eudora's matching is fast and sloppy; highly repetitive data streams and long expect strings might fool it (for example, Eudora wouldn't see "Login:" if your server said: "LogLogin:").

Eudora follows its normal .i.Timeout;time-out process when looking for an expect string. That is, after 45 seconds Eudora asks you if you want to keep waiting or cancel the process. If you cancel, the connection process is stopped. .i.Password:Forgetting:After failed expect;.i.Password:Dialup;If you use either of the password replacement sequences (".i.\p;\p" for your POP account password or ".i.\P;\P" for your dialup password) in the same string as an expect, Eudora assumes the password is wrong if the expect string isn't found. This causes Eudora to ask for your password the next time it tries to connect.

Connecting to Servers

.i.Connecting to servers:with dialup;Once the navigation is done, Eudora issues a command to connect to the proper port of the server you are using. Once this command is sent, the connection MUST be transparent, as discussed above; no echoes and no carriage return translation. Eudora comes configured to send the command:

```
.i.telnet;telnet hostname portnumber .i./stream;/stream<return>
```

The "<return>" means a carriage return in this document only; you must type actual carriage returns in ResEdit. A template for the command is kept in .i.7400.13;7400.13 ("7400.13" is shorthand for STR# resource id 7400, string 13). The template begins life as "telnet %p %d /stream\n". The %p is replaced with the hostname and the %d with the port number. It is acceptable to change this string however you please, except that %d, if it is used, must come after %p. The best way to change this string is not to modify it, but to override it. You can do this by creating an 'STR' resource of id 7413 and putting the string you want Eudora to use in that.

A Return By Any Other Name

When Eudora is communicating with your POP or SMTP server, it's important that they agree on what constitutes a line. The Internet specification stipulates that a line ends with a .i.Carriage return;carriage return followed by a .i.Linefeed;linefeed. Most UNIX systems will "helpfully" translate carriage returns into linefeeds. If you can't get that feature turned off, it may help to make Eudora send only a carriage return, and not the carriage return/linefeed pair. You may edit .i.6000.17;6000.17 to be what you want Eudora to send at the end of a line. The best way to change this string is not to modify it, but to override it. You can do this by creating an 'STR' resource of id 6017 and putting the string you want Eudora to use in that.

An Example: Direct Connection To A UNIX Box

Here's a suggest connection method if your Macintosh has a serial line to a UNIX machine, or if there are modems on your Macintosh and UNIX machine.

1.

Install the .i.Srialpop;srialpop program on your UNIX system. Source to this is part of the Eudora distribution.

2.

Put the following strings in the ".i.Navigate In;Navigate In" resource.

1:

```
\r\r\rlogin:
```

2:

```
\u\r\rpassword:
```

3:

```
\p\r\r\rle%
```

4:

```
exec srialpop
```

3.

Put one empty string in the "Navigate Mid" resource.

Resources for dialing directly into a UNIX machine.

That's it. Srialpop takes care of the terminal settings for you.

Note:

A plug-in with these strings in it is part of the Eudora distribution. The file name is ".i.Direct UNIX Navs;Direct UNIX Navs."

Need More Options?

If Eudora's .i.Scripting;scripting doesn't cut it for you, there are a couple of alternatives. One is the Calypso connection tool. It lets you use CCL scripts for the connection/disconnection process. Calypso can be found on major Macintosh archive sites. The other is the Simon Fraser University version of Eudora, which has a powerful

built-in scripting language. This is available for anonymous ftp from ftpserver.sfu.ca, pub/mac/eudora sub directory.

[Return to Table of Contents](#)