

Appendixes E-G

Appendix E – Character Sets

Introduction

This appendix discusses in detail how Eudora handles character sets and character set transliteration.

Terminology

Before discussing how Eudora handles character sets, there are some terms that need to be defined.

A character is a basic unit of written language; a letter, number, punctuation mark (or in some languages, a whole word or phrase). Major modifications to a letter (for example, capitalization or the addition of an accent mark) make that letter a separate character unto itself. “A”, “a”, “à”, and “á” are all different characters, as are “B”, “0”, “.”, and so on.

A character code is a number that is used to represent a given character. Since computers really work only with numbers, character codes are required to allow computers to deal with letters, words, and even user manuals.

A character set is a group of characters and their character codes. For example, we might decide to base a character set on the English alphabet, and simply number the capital letters from 1 to 26:

A Simple Character Set

Now, if we wanted to spell “CAT”, we’d use the numbers 3, 1, and 20.

The “US-ASCII” Character Set

The character set described above is a simple one. Too simple, in fact. What if you want to spell “The cat sat on the mat.”? You can’t, because there are only capital letters and no space or period. A long time ago, a character set was devised to fit much common United States English usage. This character set has come to be known as “US-ASCII.” It is considerably richer than just capital letters:

The US-ASCII Character Set

Using US-ASCII, you can write “The cat sat on the mat.”, using this sequence of numbers: 84, 104, 101, 32, 99, 97, 116, 32, 115, 97, 116, 32, 111, 110, 32, 116, 104, 101, 109, 97, 116, 46.

The US-ASCII character set is the one in widespread use on the Internet. Most Internet programs assume you are using it, and most Internet programs don’t support any other. However, what if you want to write “André sat on the mat.”? There is no character code in US-ASCII for “é”; so how do you tell the computer what you mean?

The Macintosh Character Set

The Macintosh allows us to describe our friend André’s perching habits. The most common Macintosh character set has a character code for “é”, as well as many other non-US characters.

The Macintosh Character Set

As you can see, the Macintosh character set is much larger than US-ASCII. In fact, it’s twice as large. The first half (character codes from 0 to 127) of the Macintosh character set is the same as US-ASCII. However, there are another 128 characters, with character codes from 128 to 255.

So, using the Macintosh character set, we can write “André sat on the mat.”, because there is a character code for “é”, 142.

The ISO Latin-1 Character Set

Unfortunately, not everyone uses a Macintosh, so not everyone has access to the Macintosh character set. The character sets that other computers use vary greatly. Most of them use character sets that are the same as US-ASCII for character codes from 0 to 127. However, if they provide characters beyond US-ASCII, they often do so with character codes other than the ones chosen by the Macintosh. That is, on some computers “é” doesn’t have a character code of 142, but might instead have a character code of 237. So, if they sent you some text with “André” in it, it would come out on your screen as “Andrè”, which would not be terribly effective.

In order to solve this sort of problem, some standard character sets have been agreed to. One popular character set is called “ISO Latin-1,” or “ISO-8859-1.”

The ISO Latin-1 Character Set

One computer can tell another “Let’s use ISO Latin-1,” and then both computers will know that the character code for “é” is 233 when they’re talking to each other, even though one may usually use 142, and the other might usually use 237.

Quoted-Printable Encoding

There is, however, a problem with using the ISO Latin-1 character set. SMTP (the protocol used to move mail around the Internet) cannot use character codes greater than 128. So our beautiful “é”, with its character code of

233, cannot be sent over the Internet. If you try, chances are it will get 128 subtracted from its value, making it 105, which is "i". "André" becomes "Andri", which just won't do.

This problem is avoided by the use of "quoted-printable" encoding. To represent a character using quoted-printable encoding, your mailer converts the value of the character to two hexadecimal digits and precedes them with an equals sign. So, "é" becomes "=E9" while your mail is being sent. Your recipient's mailer then changes the "=E9" back into an "é" and:

«Il est démontré, disait-il, que les choses ne peuvent être autrement;
car tout étant fait pour une fin, tout est nécessairement pour la
meilleure fin.» -- Voltaire, "Candide"

Quoted-printable encoding is a wonderful thing when it works. The problem is that not all mailers are as forward-thinking as Eudora, and they do not all support MIME. If your recipient doesn't have MIME, they may find the presence of quoted-printable encoding to be more objectionable than the mangling of a few special characters. They may wish they could get "André", but if they can't, they might rather have "Andri" than "Andr=E9". Also, if quoted-printable encoding is used, it affects more than just international characters. Since "=" is used in the encoding, it must be encoded specially, and all the equals signs in your mail will be turned into "=3D" while your mail is sent. Moreover, mail encoded in quoted-printable must have lines no more than 76 characters long; lines longer than that will be split in two, and an equals sign placed at the end of the first line. All this damage gets repaired if the recipient has a MIME mailer, but if they don't, it can be quite unpleasant.

Disabling Quoted-Printable Encoding

If your recipient doesn't have a MIME mailer, there are several ways to avoid using quoted-printable encoding. These are described below.

Don't Use International Characters

The simplest way to avoid quoted-printable is to not use any international characters. Avoid "André", and Eudora won't use quoted-printable. However, there is a catch to this; when Eudora sends plain text attachments and the Always include Macintosh information is off, Eudora will always use quoted-printable encoding for the attachment. This is because Eudora has to decide whether or not to use quoted-printable before it begins sending the attachment, when it doesn't yet know if the attachment contains special characters. Eudora errs on the side of caution, and always uses quoted-printable for plain text attachments.

Use Fix Curly Quotes

The Fix Curly Quotes option is a way to avoid using quoted-printable if your mail contains just a few select special characters; namely the "curly quotes" (""), bullet (•), and em and en dashes (—). Since these characters often appear in Macintosh documents, but have very reasonable US-ASCII equivalents, some users choose to have these characters changed into US-ASCII. If you turn Fix Curly Quotes on, these characters will be changed into US-ASCII, and they won't invoke quoted-printable.

Use the US-ASCII Transliteration Table

Another way to avoid quoted-printable is to install EudoraTables and choose the US-ASCII transliteration table (see the "Transliteration Tables" section). This maps all international characters to their nearest US-ASCII equivalents. "André" will become "Andre"; not great, but perhaps better than "Andri" or "Andr=E9".

Turn Off the QP Icon

The QP icon on the icon bar of a composition window controls whether or not Eudora is allowed to use the quoted-printable encoding. If you uncheck the QP icon, Eudora won't use quoted-printable for that message, no matter what.

Turn Off the May Use Quoted-Printable Option

The May Use Quoted-Printable switch in the Settings dialog (Sending Mail) controls the default setting of the QP icon. If you turn this switch off, messages you create will never use quoted-printable encoding.

Transliteration Tables

When Eudora sends mail that includes characters like "é", it normally "transliterates" them (Eudora changes the character code from the Macintosh character set to the ISO Latin-1 character set). So, "é" gets changed from 142 (the Macintosh character code) to 233 (the ISO Latin-1 character code). When Eudora receives mail, the reverse is done, and 233 becomes 142.

This process is controlled by "transliteration tables" ("tables" for short) which are stored as 'taBL' resources. A table consists of 256 numbers. Tables are used by using the character code to be transliterated as an index into the table, and replacing it with the value found at that position in the table. For example, when transliterating an "é" from the Macintosh character set to ISO Latin-1, we look at place 142 in the table (142 is the Macintosh character code for "é"); there we find a 233 (the ISO Latin-1 character code for "é"), and so we replace 142 with 233.

Eudora comes with four 'taBL' resources. Their resource id's and purposes are:

1001

ISO Latin-1 to Macintosh. This table is used to transliterate from character codes in ISO Latin-1 to character codes in the Macintosh character set.

1002

Macintosh to ISO Latin-1. This table is used to transliterate from the Macintosh character set to the ISO Latin-1

character set.

1003

Identity table. This table is provided as a reference for people who wish to write their own tables.

1004

Fix curly quotes table. This table is used by the Fix Curly Quotes switch, for people who would rather stick to US-ASCII where possible.

More Tables

If ISO-Latin-1 is not the character set for you, it is possible to get Eudora to offer you more choices. Simply drag the "Eudora Tables" document into your Preferences Folder:

Installing the Eudora Tables document

Once Eudora Tables has been installed, launch Eudora. The "Priority" menus on incoming and outgoing mail now have some new choices. These choices allow you to control how your mail is transliterated.

Priority Menus with Transliteration Tables

Incoming Messages

The table (if any) that is being used to display the current message is checked. The table that is used by default (if any) to view messages is outlined.

To change the table that is used to display a message, select the table you want to use from the Priority popup menu. The message is redisplayed using that table, and that table is used to display the message from then on.

Outgoing Messages

The table (if any) that is used when the current message is sent is checked. The table that is used by default (if any) when sending messages is outlined.

To change the table that is used to send the message, simply select the table you want to use from the Priority popup menu.

Default Tables

If you usually want to view or print your mail with a particular table, hold down the [shift] key when selecting the table from the Priority popup menu for an incoming message. The table title is outlined in the Priority popup menu to show that it is the default table, and from then on your messages are viewed with that table, unless you specify otherwise.

Note:

If an incoming message uses MIME and Eudora knows the character set the message uses, the message is transliterated before it is stored, and a viewing table is not needed or used.

If you usually want to use a particular table for outgoing mail, hold down the [shift] key when selecting the table from the Priority popup menu for an outgoing message. The table title is outlined in the Priority popup menu to show that it is the default table, and from then on your messages are sent using that table, unless you specify otherwise.

To clear the default table, hold down the [shift] key and select the outlined table from the appropriate menu. The default then becomes no table.

No Table At All

If you want a particular message not to be displayed (or sent) with any table, pull down the Priority popup menu. The table in effect for that particular message is checked. Choose the checked item; the check mark is erased and no table is used when that message is displayed (or sent).

Summaries

For non-MIME mail, the sender and subject lines are run through the default viewing table when mail arrives, and placed in the message summary (for display in mailbox windows and in the editable subject area). Subsequent viewing table changes won't affect the summaries. For incoming MIME mail, no such transliteration is done, because MIME has a mechanism for specifying character sets in names and subjects.

Ph and Finger

Ph and finger queries are transliterated according to the tables chosen at the bottom of the window:

Controlling transliteration in the Ph window

What you type is transliterated with the "Query Table," and the server's response is transliterated with the "Result Table."

Attachments

Transliteration tables are normally not used when sending or receiving attachments, unless those attachments are plain text documents. If the attachments are plain text documents, they will be transliterated if the "Always As Documents" option is turned off, or if the "AppleDouble" attachment type is chosen.

Creating New Tables

If you are trying to use a character set that Eudora doesn't understand, you can build tables for it. You will need to create two 'taBL' resources, and probably your own 'euTM' resource as well.

Choosing Resource Id's

You need to choose two resource id's for your tables. These id's should be consecutive, with the lower-numbered id being odd. The odd-numbered id is used for incoming mail, and the even-numbered table is used for outgoing mail.

In order to avoid id conflicts, take the Macintosh country code, multiply by 10, add 2000, and add 1 if the table is for incoming mail, or 2 if the table is for outgoing mail. For example, the table that maps Swedish ASCII to Macintosh characters is:

10*7 (seven is the country code for Sweden) + 2000 + 1 (since the table is used for receiving mail), or 2071.

Creating the 'taBL' Resources

Once you've chosen id's, make the 'taBL' resources. ResEdit's general editor works quite well for tables. You will probably wish to copy the 'taBL' resource id 1003 to serve as a starting point. That way, you only need modify the parts of the Macintosh character set that need to be transliterated. The names of the resources will be used in the menus, so name the table resources descriptively. It's also a good idea to create your resources in a "plug-in" file; a file with type 'rsrc' and creator 'CSOm'. That way, users can easily install and remove your table, and your table won't get wiped out if they upgrade their copy of Eudora or EudoraTables.

Creating an euTM

The 'euTM' resource is used for naming character sets. Character sets must be named so that mailers know which character set is being used. The official MIME names for character sets are often very unpleasant. For example, the name for a common Swedish character set is "SEN_850200_B."

Part of an euTM Resource

The 'euTM' resource is a list of resource id's and names. When Eudora is sending mail, it will subtract 1 from the table's resource id, then look for that resource id in all the 'euTM' resources it can find. When it finds a matching id, the name corresponding to the id is used.

For example, a user choosing the Mac->se table would be using table id 2072. Eudora subtracts one, finds 2071 in the second position in the 'euTM' resource, and sends the mail with a character set name of "SEN_850200_B."

When receiving mail, the process is reversed; the character set name is looked up, the resource id found, and that transliteration table used for the mail.

For your table, you should create an 'euTM' resource, list the resource id of your table (only the odd id), and the name that should be used in mail for the character set.

Appendix F – Using UUCP

Introduction

.i.UUCP;Eudora works with UUCP in almost exactly the same way as it works with the POP and SMTP servers.

.i.Attachments:And UUCP;Attachments are supported, as is regular mail checking and the other features. It is possible to mix methods; for example, you can use UUCP for reading mail but SMTP for sending it.

Eudora does not come with UUCP. Three available Macintosh UUCP systems are ".i.uupc;uupc 3.0"

(dplatt@snulbug.mtview.ca.us), ".i.gnuucp;gnuucp" (jim@fpr.com) and ".i.UUCP/Connect;UUCP/Connect" (formerly "µAccess," sales@intercon.com). ".i.µAccess;UUCP/Connect" is commercial; the other two are free ware. Eudora has been tested with all three packages; it works well with uupc 3.0 and UUCP/Connect, but it does not work very smoothly with gnuucp.

Settings Dialog for UUCP

Personal Information settings for UUCP

Hosts settings for UUCP

POP Account

.i.POP Account:And UUCP;If you are going to receive mail via UUCP, you should put the full path name of your ".i.Mail drop;mail drop" (the file where UUCP leaves mail for you) in the POP Account field. Precede the name with an exclamation point.

MacTCP/Communications Toolbox

This setting doesn't matter if you're doing pure UUCP mail. If you're trying to mix UUCP with SMTP or POP, set this to whatever is appropriate for your SMTP or POP connection.

SMTP Server

.i.SMTP Server:And UUCP;If you want to send mail via UUCP, several items have to go in the SMTP Server field.

Each one should be preceded by an exclamation point. The items are, in order:

mac - the UUCP name of your Macintosh.

spoolpath: - the full path name of the UUCP working directory.

user - your user name on your mac

0000 - a four-digit sequence number; will be incremented by Eudora.

Return Address

.i.Return Address:And UUCP;If you use UUCP for reading your mail, you must put your correct return address in the Return Address field. It is absolutely vital that this address be correct. If it's wrong, no one is able to reply to your

mail and the mail transport system is unable to tell you your mail can't be delivered.

Operation

Almost all Eudora features work normally with UUCP. The one exception is the .i.Leave Mail on Server option:And UUCP;Leave Mail On Server (LMOS) option. When Eudora is used with POP, setting LMOS results in Eudora downloading only unread mail. With UUCP, however, Eudora does not distinguish between read and unread mail; it downloads all the mail at each check. This results in duplicate messages, unless you use some other means to clean out your mail drop between Eudora checks. It is suggested that the LMOS option remain turned off when using UUCP.

Internals

Mail Drop Format

.i.Maildrop:UUCP:Format;Eudora expects the mail drop to be in standard UNIX mailbox format, with UUCP envelopes ("From " lines) at the beginning of each message. The mail drop should use carriage returns (not line feeds) for new lines.

Working Files

.i.UUCP:Working Files;When sending mail, Eudora creates two files in the UUCP work directory. These files are:

D.mac0####

.i.UUCP:Working files:D.relay0####;The message itself is put in this file. As distributed, Eudora uses returns for new lines in this file. That can be changed by editing the last characters of STR# resource id 8000, string 5; Eudora will use whatever non-printable characters are at the end of the string. The mail begins with a UUCP envelope. The "####" stands for the four-digit sequence number mentioned in the "SMTP Server" section above. It increments as each message is sent.

X.mac0####

.i.UUCP:Working files:D.mac0####;Commands for the UUCP system are put in this file. These commands are all editable via ResEdit; they are distributed with returns at the ends.

U user mac

; identifies you (STR# id 8000, string 1)

F D.mac0####

; this file contains your message (8000,2)

I D.mac0####

; use your mail for input (8000,3)

C rmail recipient...

; all recipients of the mail are listed here (8000,4)

Appendix G – MIME and Mapping

What is MIME?

"MIME" stands for Multipurpose Internet Mail Extensions. MIME serves two major purposes – it allows mail applications to tell one another what sort of data is in mail, and it also provides standard ways for mail applications to encode data so that it can be sent through the Internet mail system.

MIME Encodings

The Internet uses the "SMTP" protocol to move mail around. SMTP is limited to the US-ASCII character set (see Appendix E). This is a problem for people who speak languages other than American English and so need accented characters or non-American letters, or for people who want to use special symbols like section mark (§).

MIME provides a way around this restriction. It offers two encodings, "quoted-printable" and "base64." These encodings use US-ASCII character codes to represent any sort of data you like, including special characters or even non-text data.

"Quoted-printable" is used for data that is mostly text, but has special characters or very long lines. It's very simple. Quoted-printable looks just like regular text, except when a special character is used. The special character is replaced with an "=" and two more characters that represent the character code of the special character. So, a section mark (§) in quoted-printable looks like "=A8".

However, there are some other things that quoted-printable does. For one, since it uses an "=" to mean something special, equal signs must themselves be encoded (as "=3D"). Second, no line in quoted-printable is allowed to be more than 76 characters long. If your mail has a line longer than 76 characters, the quoted-printable encoding will break your line in two and put an "=" at the end of the first line, to signal to the mail reader at the other end that the two lines are really supposed to be all one line. Finally, a few mail systems either add or remove spaces from the

ends of lines. So, in quoted-printable, any space at the end of a line gets encoded (as “=20”), to protect it from such mail systems.

Let's try an example. Here's a passage of text that you might type on your Macintosh:

«Il est démontré, disait-il, que les choses ne peuvent être autrement; car tout étant fait pour une fin, tout est nécessairement pour la meilleure fin.»

Without any encoding, this might show up on your recipient's screen as:

+Il est dimontri, disait-il, que les choses ne peuvent btre autrement; car tout itant fait pour une fin, tout est nicensairement pour la meillure fin.;

This corruption happens because SMTP cannot handle the special characters. However, if you and your recipient both have MIME, quoted-printable encoding would be used, and your text would show up properly:

«Il est démontré, disait-il, que les choses ne peuvent être autrement; car tout étant fait pour une fin, tout est nécessairement pour la meilleure fin.»

While your mail was actually in transit, however, it would have looked like:

=ABII est d=E9montr=E9, disait-il, que les choses ne peuvent =EAtre =
autrement; car tout =E9tant fait pour une fin, tout est n=E9cessairement =
pour la meillure fin.=BB

Base64 encoding is another way to protect binary data from the SMTP mail system. However, Base64 makes no attempt to be legible, and is most appropriate for non-text data.

MIME Labelling

The other important part of MIME is that it lets mailers communicate what kind of data is in a message (or part of a message). The primary mechanism used for this is the Content-Type header:

Content-Type: text/plain; charset=iso-8859-1

A content-type header is divided into three parts; the content type, the content subtype, and the parameters. In this case, the content type is “text,” meaning the message contains mostly legible text. The content subtype is “plain,” which means there aren't any formatting commands or anything like that embedded in the text. Finally, “charset=iso-8859-1” is a parameter; in this case it identifies the character set the message uses.

The major content types are:

text

legible text

image

pictures and graphics

audio

sound

video

moving pictures

message

messages or pieces of messages

multipart

several different kinds of data in a single message

Practical Issues

There are really only two things you sometimes need to do with Eudora and MIME. One is that it may occasionally be necessary to turn off quoted-printable encoding. Another is that you may want to know how to define mappings between MIME types and Macintosh types.

Turning Off Quoted-Printable

Eudora automatically uses quoted-printable encoding if your mail contains special characters. Eudora also uses quoted-printable encoding for attached plain text files. If your recipients don't have MIME, quoted-printable may hurt more than it helps. If that's the case, just turn off the QP icon when you are sending text files to those recipients.

Turning Off Quoted-Printable Encoding

Mapping Between MIME Types and Macintosh Types

When you send attached files to other Eudora users, Eudora automatically knows what kind of data is in the files, because Eudora sends along special information with the file. However, if you're sending the file to a non-Macintosh user, or receiving files from a non-Macintosh user, it's important to get the right MIME type information on the file, or for Eudora to understand what the MIME type information means.

Eudora knows about some MIME types. However, since new MIME types are being defined all the time, it may be necessary to add to Eudora's knowledge from time to time. If you're familiar with ResEdit, this isn't too hard to do. The way Eudora maps between MIME and Macintosh types is with EuIM and EuOM resources. EuOM resources are used for sending attachments, EuIM for receiving. They have the same basic structure. EuOM and EuIM resources are lists of individual elements called "maps." Each map describes a Macintosh document type (or MIME data type) and then lists what MIME data type (or Macintosh document type) it corresponds to. For any given type, Eudora looks through all the maps in all the EuOM or EuIM resources, and uses the best match.

Note:

EuOM and EuIM resources are also used when uuencoding and uudecoding files, so that filename suffixes can be mapped to and from Macintosh types. A good set of EuIM and EuOM resources can substantially improve document exchange with systems that use uuencode.

Sending

When you create a map in an EuOM resource, you use the "Creator Code" and "Type" fields to specify what documents the map applies to. These fields should be filled with the four-byte creator code or Macintosh type of the documents you want to send. If you leave the Creator Code blank, but fill in the type, the map is used for any document of that type, regardless of creator. If you fill in both Creator Code and Type, a document has to match both for the map to be used. Given the choice, Eudora uses the map that matches both creator and type.

The other parts of the map are used to construct the MIME information. Content Type and Content Subtype are the MIME type and subtype to use for the document. Filename suffix allows you to tell Eudora to add a suffix to the filename, as an extra hint to the receiving system (for example, you might have Eudora add ".xls" to Excel files). "Newline conversion?" tells Eudora whether or not to convert carriage returns in the file to carriage return, linefeed. Usually, you should set this to 1 for text data, but to 0 for binary files.

Finally, "May suppress resource fork?" is used in conjunction with Eudora's Always include Macintosh information. If you set this to 1, and Always As Documents is off, Eudora won't send Macintosh type and creator information with the file, and won't send the resource fork. Instead, it will just send the data fork with the MIME information attached to it.

An Example Map in an EuOM Resource

The map above says that all files of type "EPSF," no matter what the creator, should be sent as "application/postscript," that ".eps" should be added to the filename, that carriage returns should not be turned into carriage return/linefeed pairs, and that when the Always include Macintosh information is off, the resource fork won't be sent.

Receiving

EuIM resources are used for receiving files. They're pretty much the same as EuOM resources, except that the MIME type and subtype are used for matching, and the Macintosh creator code and type are applied to the file received.

As with EuOM resources, you can leave parts blank. If you want to match all files with an ".eps" suffix, regardless of the MIME type or subtype, leave the type and subtype blank. If you don't care what the filename suffix is, leave that blank and match with the MIME type and/or subtype only. Again, as with EuOM resources, Eudora will choose the map that matches best.

With EuIM resources, it's sometimes a good idea to use several maps to catch all important cases. For example, it might be a good idea to have three maps for dealing with PostScript files, as follows:

Content Type: application

Content Subtype: postscript

Filename suffix:

Creator Code: mlpr

Type: TEXT

This map will catch most MIME PostScript files, and set their creator to MacLPR.

Content Type:

Content Subtype:

Filename suffix: .eps

Creator Code: dPro

Type: EPSF

This map will match any incoming file with a suffix of ".eps," regardless of the MIME type info, and set its type to "EPSF" and creator to "dPro" (MacDraw Pro). But what if a file comes in with a suffix of ".eps" and a MIME type/subtype of "application/postscript"? Which map gets used? The first one gets used; when Eudora has a choice between matching a suffix and matching MIME type information, MIME wins. A third map may be in order:

Content Type: application

Content Subtype: postscript

Filename suffix: .eps

Creator Code: dPro

Type: EPSF

This makes application/postscript files with suffixes of “.eps” get type EPSF and creator dPro.

[Return to Table of Contents](#)