A PARTICULAR VIEW OF EUDORA CLASSICAL AND ITS SOLUTIONS TO CRITICAL PROBLEMS OF OBSOLETENESS

February 28, 2026

INDEX

INTRODUCTION

This document was originally written in Spanish. The English version is a machine translation that was subsequently reviewed by a non-professional and non-exhaustive human.

The publication of this document has an urgent component, because there are people in forums expressing problems, which I take to be quite distressing, in their use of Eudora, due to the critical problems of classic Eudora (versions 3 to 7) due to its obsolescence, such as,

above all; the problems caused by the lack of secure protocols in classic Eudora that would allow it to transfer mail with ISP servers, and on the other hand; the coexistence with "Mojibake" which causes incompatibility with UTF-8.

One might think that the major problem with secure protocols should have surfaced long ago and that a solution would have been settled long ago. But we still see users (I've seen some this week alone) whose ISP or other systems have changed something that prevents them from exchanging email due to TLS protocol issues, etc.

Perhaps the experience described here, e.g., the use of MTAs or Stunnel, will offer quick solutions to those people.

Therefore, consider publishing immediately even if other documentation might be useful or necessary (e.g., on the detailed use of the tools provided). On the other hand, it's also possible that this documentation won't be requested, making its production pointless.

EMAIL AND EUDORA

Email is a medium that has always been inseparable from the Internet because, even before the medium itself in the extent that the Internet means, it offered very powerful and pioneering communication features.

The first and most fundamental advantage is that it allows communication without requiring participants, who can be numerous, to be online simultaneously, as is the case with telephone communication. It thus constitutes **unattended communication, yet as agile as desired** , simply by checking email as frequently and at any time.

This is a truly **disruptive technology** that, in addition to exchanging messages, has the capacity to exchange documents, images, and files in general. And because it uses the Internet (originally ARPANET, with the first email sent in 1971) as its host or "mother medium," it is a virtually instantaneous means of communication for the entire world and beyond, and, as a bonus, extremely cheap, especially once the level of Internet implementation was achieved, which was reached quickly.

Email **solved a critical need for progress** , and after 50 years of existence, it maintains its functionality with very abundant use.

Therefore, it is logical that many people are interested in having an email environment in which they feel functionally satisfied and emotionally comfortable.

Although web interfaces appeared years after the implementation of email (the ability to keep email on a server, without downloading it to a local computer, and manage it against the server through a web interface), and although this option seems to be gaining users, I think that this is due to a series of rather circumstantial reasons, and that if the dilemma is subjected to even a superficial analysis, the conclusion is that it is better to manage email as it was initially done for many years: with a client application or program from the servers.

I'm one of those who discovered Eudora many years ago, decades even, and from the very beginning, we found it possessed a wealth of ideal features that remain relevant today. As you can see in the "Eudora users" forums, we haven't found anything better after all these decades, and there are quite a few of us. We continue to use the classic Eudora even given the obsolescence that each of us has navigated in various ways.

WHERE IS THE SUITABILITY OF EUDORA CLASSICO?

From my point of view; basically in the way of being able to have everything necessary and convenient regarding email, **in sight and at hand** .

While this "whole" isn't exactly perfect—because nothing is, and because it's subject to progress and its evolving requirements—for many, it effectively constitutes a complete system in practice, one that hasn't been surpassed by other email solutions. I'll illustrate this with a few specific features as examples:

* The way to make multiple accounts or "personalities" available to the user.

* The same applies to mailboxes or trays other than those used for mail, and their contents.

* A highly configurable and adaptable user interface, with the possibility of multiple windows, floating or fixed, adaptable and adjustable to each other.

* A very powerful and efficient (very fast) internal search engine.

* The design of the configurations and functionalities (filters, etc., etc.)

* The ability to easily view complete message headers.

* The templates ("Stationery").

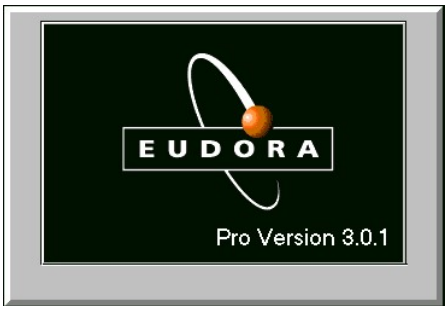
Etc.

And although it wasn't a goal at the time, it's now very interesting for security and privacy reasons: the fact that Eudora Classic can be configured to open emails with a **viewer. (HTML code interpreter)** minimum ("Use Microsoft's The viewer (Y/N) interprets only the bare minimum of the message, allowing the text to be seen and little else, but preventing the execution of instructions that could compromise security and privacy. It did implement, and was conceived as a security measure, the ability to allow or disallow the execution of executables contained within the message. However, without dealing with actual executables, the minimal viewer prevents the execution of HTML instructions that are truly executable.

In short; for many, including myself: Eudora Classic was and is a **success in its design**, unrivaled by other email clients, and even less so by web interfaces.

MY SPECIFIC EXPERIENCE

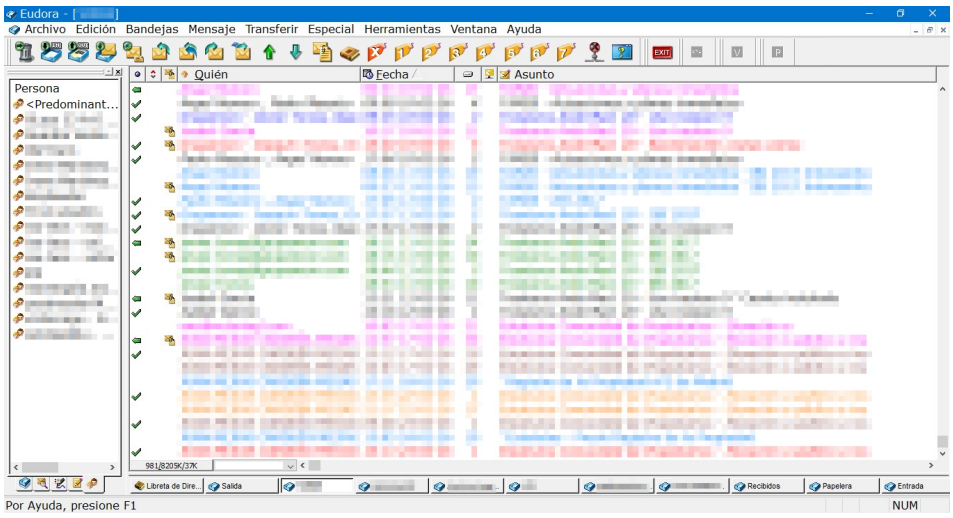
Eudora was the first email client program I ever used. It was version 3. It must have been around 1996. From shortly after, I still have the folders called Disk1 to Disk4 from a version 3.0.1 of Eudora:



And I kept updating until I had version 6.1.2.0 translated into Spanish, which is the one I've been using for about 30 years:

The Spanish translation, done by someone who identifies as " GRB " (probably their initials), and who, judging by the flag, is from Uruguay, must have been an incredibly difficult task to patch an executable for which they didn't have the source code (a heroic effort). And what's more, as a Spanish translation, it's very faithful to the original Spanish. Congratulations and thank you!

A picture of my general "setup":



Why haven't I updated to v7.1.0.9?. Because I believed at the time that, aside from a few interesting but not crucial features, like having a good set of emoticons, etc., the biggest improvement between v6 and v7 was an even faster search engine (I think it's called X1), which could also be disabled, thinking about searches across tens of thousands of emails. And since, in my case, even with thousands but not tens of thousands of emails, searches in v6 are impressively fast, I preferred a conservative approach, especially since I only found out

about v7 when it was already known that Qualcomm was discontinuing the product, and the situation made me uncertain.

By the way, I've never had any compatibility issues with Windows. Currently, and for years now, my Eudora runs on Windows 10 x64 . And let me clarify, I'm not a Windows fan.

AND THEN CAME THE PROBLEMS IN THIS GREAT STORY

Qualcomm's Eudora (classic Eudora) reached version 7.1.0.9 in 2006 (October 11, 2006), almost twenty years ago, and Qualcomm decided not to continue with the product.

Qualcomm considered creating an "Eudora OSE" (Open Source Edition) version based on Thunderbird . However, the project's focus shifted and ultimately failed.

There are details at this point and in the subsequent story that are exposed on websites and that, above all, Eudora's followers know, with one of the milestones being the deposit of the historical source codes for public knowledge in the Computer History Museum (CHM). Although not all the source code was deposited, because there were parts of Eudora that were not developed by Qualcomm, but used from external libraries, for which Qualcomm did not have a license to provide the source code, and this is a key point for the current continuity problem.

CRITICAL PROBLEMS WITH EUDORA'S OUTDATING. HOW HAVE I SURVIVED THEM?

Eudora has two critical outdated problems, and here I describe how I have managed them:

1) SSL/TLS.

Security protocols evolved critically for the continuity of a maintenance-free Eudora system, because general evolution brings with it the emergence of critical security threats, which necessitated these protocols. This was due, on the one hand, to internal security concerns, and on the other, because Internet Service Providers (ISPs)

mandated their implementation: if you don't comply with these protocols, you won't receive email.

Since I've recently started looking for Eudora enthusiasts online again, I'm seeing and being amazed at how people (this bunch of Eudora fanatics, like me) have survived this and other inconveniences.

The simplest and most straightforward approach has been that of people who started using Stunnel as an intermediary between their Eudora and the ISPs' servers. Logical, simple, fully functional, and therefore; straightforward.

In my case, I didn't look for the solution; it came to me. It arrived practically at the same time that it began to manifest itself as a need.

Even before the need arose, I knew I wanted an MTA on my machine (for a while they've been colloquially called "SMTPs" but that's incorrect and is being corrected: they're Mail Transfer Agents; MTAs). MTAs are software programs developed to handle email transfer between servers; to serve clients whose internet-facing accounts reside on those servers. Basically, where you connect to download your email. MTAs handle the conversations, using different protocols (POP, SMTP , etc.), with other servers where other users' accounts are located, and with intermediate servers that simply transfer email. These conversations are recorded, generating precise logs of what happened with each message.

For many years I've wanted to control my email traffic and message traces, and even before the aforementioned needs arose, I already had an MTA "on my machine". This means my Eudora doesn't connect to ISP servers, as most users do, but to the MTA server on my own machine. And because the most efficient way for both parties to communicate (on the same machine) is without security protocols (plain text), my Eudora has always been able, and still is able, to avoid using secure protocols (SSL/TLS) since the MTA, which handles communication with internet servers, already uses them.

I've switched MTAs a couple of times, but it's been seamless for my Eudora; it hasn't even noticed. For the last few years, I've been using Mercury Server as my MTA, for which I've developed a Plugin.

2) UTF-8.

Eudora only understands the Western ANSI (Windows-1252) / ASCII character encoding. When UTF-8 encoding appeared and was

implemented, Eudora began displaying strange characters in received emails.

Result: the vast majority of incoming emails have characters, sometimes many, replaced by other "weird" ones, forming a rarefied and unfriendly text, known as "Mojibake".

How have I survived this?

Especially with a Word macro I created that corrects the mojibake. I copy the mojibake infused body text into a blank Word document (I always keep a blank Word document open for various uses), trigger the macro with a key combination, and copy the result back into the email body (and save it corrected, of course). I used to handle the Subject line manually in its own window (the TOC content for the techies).

However, I have recently developed some "Widgets" (software tools) for Eudora: a Plugin and an external Toolbox, the latter of which I have called EudoraToolbox, which solve that and other problems for me (don't worry, they are freely available to anyone who wants them. Just so you know: they are made for me).

REST OF THE OVERVIEW: OTHER POINTS OF OUTDATING AND IMPROVEMENT OF EUDORA.

To complete the picture, here is my full list of problems and needs for updating Eudora:

CRITICAL NEEDS

1) SSL/TLS

2) UTF-8. Eudora only understands Western ANSI (Windows-1252) / ASCII.

OF VERY HIGH INTEREST

3) New HTML viewer (email HTML interpreter). With the option to choose and also switch to a rudimentary version.

4) New email editor. With many more features such as a text editor and HTML editor.

OF HIGH INTEREST

5) TOC / MBX System -> DB.

INTERESTING

6) An even more powerful search engine (although this has always been, and still is, one of Eudora's strengths), e.g., being able to perform SQL queries on both TOC and MBX content .

7) Down Tabs (the arrangement of tray tabs at the bottom of the screen).

To control and be able to set the order. And to be able to use colors like in Excel spreadsheets.

8) Improved management of email attachments.

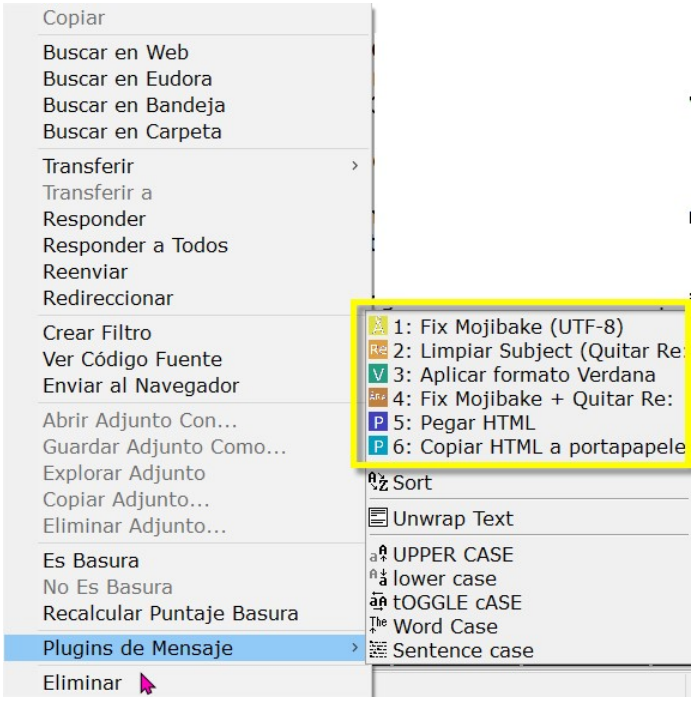
MY WIDGETS

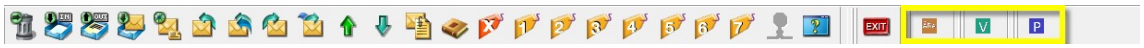
They try to solve and facilitate a few things.

Why an internal plugin and an external toolbox for Eudora?

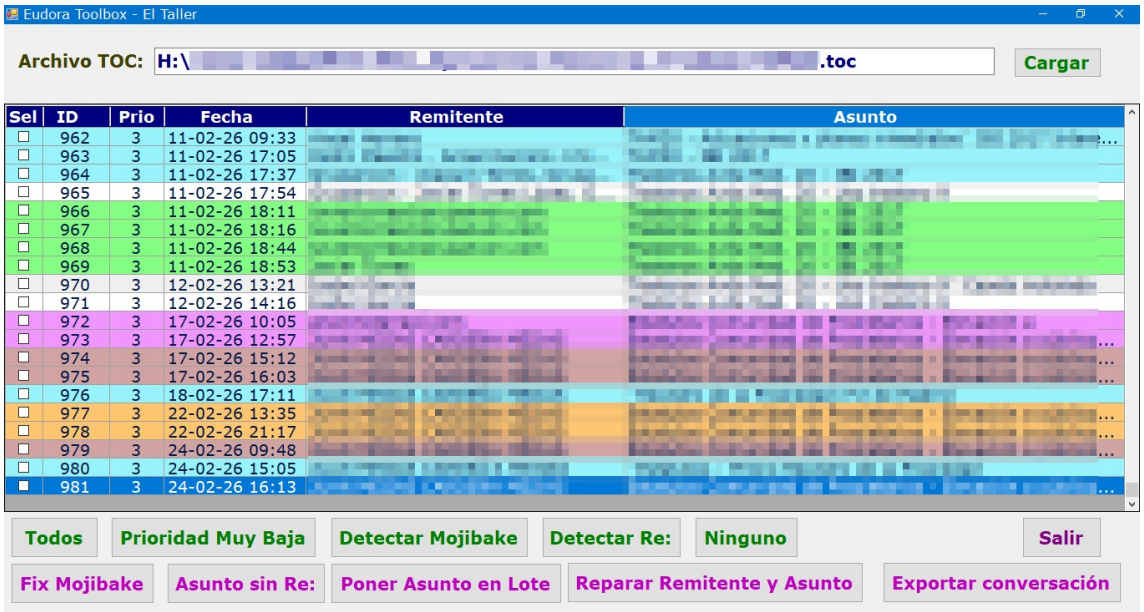
As any Eudora user with a bit of technical knowledge knows, the files that store message information are of two types: MBX and TOC. Ideally, everything should be done internally within Eudora, using plugins, and in fact, this is recommended for anything affecting MBX files because Eudora's engine modifies them correctly and safely. However, doing this from outside is complex. But with TOC files, where header information for each message is stored (date(s), subject, sender, etc.), I haven't found an efficient way to do it from within Eudora because Eudora loads TOC contents into RAM. Any modifications made with open messages in Eudora conflict with the data Eudora dumps from RAM, reverting the changes. There are more details, but that's the general reason.

Therefore, the solutions are distributed among the plugin (internal to Eudora):





and the Toolbox (external and which must work with Eudora closed):



At this time, as of today, they are developed specifically for me, so they only reflect my objectives, and more specifically: the button and menu labels are in Spanish (I am Spanish). Some minor font changes are also adjusted to my preference (Verdana, etc.).

However, the functionalities (what they do on MBX and TOC files) transcend the language of the interfaces; they are valid for users with other languages (except non-Western languages), as long as they have v6 / v7 versions of Eudora and perhaps earlier.

As of today, those were the objectives, and that's how things stand. Things that, as they are, I'm making available free of charge and without obligation on my part: I've been using them for a while and haven't encountered any problems, I'm referring mainly to their security, to them not breaking anything, but **I don't offer a guarantee**: you have to work with backup measures, as always, and test environments.

The Spanish labels and buttons shouldn't be a big problem for handling these "widgets" because there are very few menu options and buttons, and you can easily find out what each one does through immediate translations and testing.

More detailed information about each tool may be available in its own documentation (depending on demand). Currently, only this simple presentation and the executable files for the tools themselves are available.

Just to clarify, because they aren't immediately obvious, in the Toolbox:

* If you click twice—not double-click, but two slightly spaced clicks—on the Subject or Sender field, you'll enter the field's editing mode and be able to modify it, whether it's an incoming or outgoing email. The software will automatically truncate the text if it exceeds the field's limits.

* Besides the fact that there are several clearly visible and meaningful ways to select emails for the Toolbox to work on, the "Very Low Priority" button will select emails with very low priority. What does this mean, and why is it designed this way? It's the only way I've found to select, from within Eudora, the emails the Toolbox will work on: very low priority is probably the least used option and therefore the one best suited for this purpose. In Eudora, we select the emails, mark them with very low priority, close Eudora, open the corresponding TOC in the Toolbox, click the "Very Low Priority" button, and only the emails we selected in Eudora will be selected. Once we've worked on them using any of the available actions, their priority will return to Normal (or level 3).

* The email colors inherit the colors of the labels assigned in Eudora. Yes; this one is in fact obvious.

As Eudora users should know by now, plugins are DLL files that are placed, with Eudora closed, in the " plugins " directory, which is located within the Eudora installation files directory (not the directory containing the data files, such as MBX , TOC , etc.). When Eudora is reopened, the plugin 's functionality will automatically appear .

CREATE A TEST ENVIRONMENT FOR EUDORA

The following deployment has no guarantee whatsoever.

A test environment for Eudora can be set up perfectly on the same machine as the production environment, with just a few precautions.

The Eudora program's installation directory (where the `Eudora.exe` executable is located) could be shared between the two work environments; production and testing, by launching it from the shortcuts with something like this:

Production: `"C:\Program Files ( x86 )\Eudora\Eudora.exe" D:\email\mypersonalmail\`

Tests: `"C:\Program Files ( x86 )\Eudora\Eudora.exe" D:\emailtests\mytests\`

But, in order to have a total sandbox environment (a totally isolated environment), and for other reasons, it's advisable to also clone the installation directory (the directory of the `Eudora.exe` executable). And it's not necessary and in fact it's not recommended to run a second installation process. Simply clone it. The shortcuts would then be:

Production: `"C:\Program Files ( x86 )\Eudora\Eudora.exe" D:\email\mypersonalmail\`

Tests: `"D:\emailtests\Eudora\Eudora.exe" D:\emailtests\mytests\`

Having cloned both directories; both the installation and data directories:

```
C:\Program Files ( x86 )\Eudora\
->
D:\emailtests\Eudora\
```

```
D:\email\mypersonalmail\
->
D:\emailtests\mytests\
```

The executables and their associated files and folders will be located in two directories (`C:\Program Files ( x86 )\Eudora\` and `D:\emailtests\Eudora\`) without requiring a second installation process. This is possible and facilitated precisely because Eudora is software from the 1990s, and in those years, software was much more autonomous and less dependent on the Windows Registry than it is today.

By doing it this way, in addition to the total isolation between the two environments, we have the advantage of avoiding, for testing purposes, directories such as `C:\ Program Files (x86)` or `C:\ Program Files`, which are directories or folders protected by the system and will easily give very inconvenient access denials when it comes to testing, and we also protect them as well.

Two final, VERY important points:

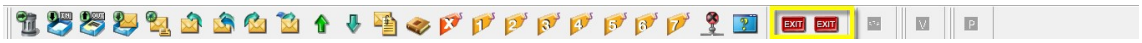
1) Review the INI file (`Eudora.ini`) located in the test data directory for any paths that point to the production data directory. Pay particular attention to the attachments directory.

`AutoReceiveAttachmentsDirectory =… \attach`

and change the path to the `\attach` directory of the test data folder.

2) It is highly advisable not to have both the test and production versions of Eudora open at the same time. It is also advisable to add something to the interface to distinguish which environment you are in.

In production, I have one exit button, and in testing I have two, and the difference is very noticeable:



CURRENT ATTEMPTS TO REVIVE EUDORA

There is the well-known initiative of the Hermes group, which managed, apparently with considerable effort, to recompile a v7.1.0.9, in what remains a mystery to me: how did they manage it

without part of the source code (those parts that Qualcomm used from third parties and did not provide the sources to CHM)?.

According to Hermes, they've been working on updating Eudora for a few years now. It seems, or at least it seems to me, that they're aiming for a commercial, non-open-source version. They provide development versions to those who donate to the project.

I see in the forums that people are understanding about the delay, which Hermes acknowledges, but they are demanding functional solutions now because they want THEIR Eudora.

My impression is that the set of requirements they have imposed on themselves is not suitable for carrying out the project; I hope I am wrong.

If I'm correct, the first problem is that a closed development group, probably small, and perhaps part-time, would have to be very efficient to generate the necessary deployment.

Starting with most of Qualcomm's Eudora source code can be a poisoned chalice if you don't make fine decisions: what do I keep, what do I add, how do I do the new things, and how do I integrate them with the old?. And all of this while changing the final product, both functionally and aesthetically; ONLY in ways that the audience strongly identified with the original Eudora can easily accept as an effective and efficient evolution, not as something different or a step backward. When something works, be very careful what you change.

There are many decisions and a lot of work involved, and I believe the best approach would be to undertake it with a fully open-source strategy, with a team as large as the volunteers who volunteer, and, of course, with coordinators who know how to make all those decisions correctly and coordinate them effectively. A similar development example would be the phpBB forum software.

And on the other hand, there are the mailing lists on Listmoms (http://www.listmoms.net); with the Eudora-Win list for users and Eudora-Dev for developers.