



Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

State of HERMES (project)

11 messages

Nicholas Edward Werner-Matavka <n.theodore.matavka.files@gmail.com>

12 June 2019 at 10:06

To: sbrothy@gmail.com, pete@maclean.com

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking. This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

Nicholas Edward Werner-Matavka <n.theodore.matavka.files@gmail.com>

12 June 2019 at 10:07

To: sbrothy@gmail.com, pete@maclean.com

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the

insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both HunsPELL and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking. This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

Pete Maclean <pete@maclean.com>

17 June 2019 at 12:09

To: Nicholas Edward Werner-Matavka <n.theodore.matavka.files@gmail.com>, sbrothy@gmail.com

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical. Which, as you say, means using the Stingray library, no matter how distasteful we find that to be. And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it. Even if that money were suddenly dropped in our laps, we might still have two significant obstacles. One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used. The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know. In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy. I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release. Specifically this is "Microsoft Visual Studio .NET 2003". I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables. The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking. This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>
To: Pete Maclean <pete@maclean.com>

20 June 2019 at 19:38

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do. They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process. I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out. Stingray 2004 is available here: https://drive.google.com/open?id=1KHzMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother. I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed. Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive. Let's save that for if we really must.

Cordially,

N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean <pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical. Which, as you say, means using the Stingray library, no matter how distasteful we find that to be. And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it. Even if that money were suddenly dropped in our laps, we might still have two significant obstacles. One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used. The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know. In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy. I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release. Specifically this is "Microsoft Visual Studio .NET 2003". I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables. The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ

unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking. This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

--
----- BEGIN TECO SIGNATURE BLOCK -----
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC\$
QA:^US\$QP:^US\$QD:^US\$QI:^US\$QA:^US\$QM:^UQ\$QG:^UQ\$QA:^UQ\$QP:^UQ\$
QE:^UQ\$QO:^UU\$QC:^UU\$QH:^UU\$QI:^UU\$QD:^UU\$QM:^UI\$QY:^UI\$QD:^UI\$
QT:^UI\$QR:^UI\$QR:^UB\$QL:^UB\$QY:^UB\$QI:^UB\$QT:^UB\$GI-5CGUGS-5CGB10CGQ0JT\$\$
----- END TECO SIGNATURE BLOCK -----
(Don't forget: ^ in TECO means just that, and \$ means press the Esc key!)

Pete Maclean <pete@maclean.com>
To: Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

21 June 2019 at 11:02

Okay, I will revert to using VS 2015 for QCSSL. That will make the job a bit easier.

Pete Maclean

At 07:38 PM 6/20/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do.Â They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process.Â I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out.Â Stingray 2004 is available here:Â https://drive.google.com/open?id=1KHZMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother. I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed. Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive. Let's save that for if we really must.

Cordially,

N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean <pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical. Which, as you say, means using the Stingray library, no matter how distasteful we find that to be. And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it. Even if that money were suddenly dropped in our laps, we might still have two significant obstacles. One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used. The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know. In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy. I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release. Specifically this is "Microsoft Visual Studio .NET 2003". I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables. The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell

and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking.Â This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing.Â I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

--
----- BEGIN TECO SIGNATURE BLOCK -----
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC\$
QA:^US\$QP:^US\$QD:^US\$QI:^US\$QA:^US\$QM:^UQ\$QG:^UQ\$QA:^UQ\$QP:^UQ\$
QE:^UQ\$QO:^UU\$QC:^UU\$QH:^UU\$QI:^UU\$QD:^UU\$QM:^UI\$QY:^UI\$QD:^UI\$
QT:^UI\$QR:^UI\$QR:^UB\$QL:^UB\$QY:^UB\$QI:^UB\$QT:^UB\$GI-5CGUGS-5CGB10CGQ0JT\$\$
----- END TECO SIGNATURE BLOCK -----
(Don't forget: ^ in TECO means just that, and \$ means press the Esc key!)

Pete Maclean <pete@maclean.com>

24 June 2019 at 15:07

To: Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

I am curious to try a build of the main Eudora executable with this Stingray 2004 package. Would you like me to do so? If yes then I need a Validation Code to install Stingray.

Pete Maclean

At 07:38 PM 6/20/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do.Â They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process.Â I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out.Â Stingray 2004 is available here:Â https://drive.google.com/open?id=1KHZMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother.Â I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed.Â Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive.Â Let's save that for if we really must.

Cordially,

N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean <pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical.Â Which, as you say, means using the Stingray library, no matter how distasteful we find that to be.Â And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it.Â Even if that money were suddenly dropped in our laps, we might still have two significant obstacles.Â One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used.Â The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know.Â In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy.Â I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release.Â Specifically this is "Microsoft Visual Studio .NET 2003".Â I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables.Â The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise.Â Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices.Â Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking.Â This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

--
----- BEGIN TECO SIGNATURE BLOCK -----
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC\$
QA:^US\$QP:^US\$QD:^US\$QI:^US\$QA:^US\$QM:^UQ\$QG:^UQ\$QA:^UQ\$QP:^UQ\$
QE:^UQ\$QO:^UU\$QC:^UU\$QH:^UU\$QI:^UU\$QD:^UU\$QM:^UI\$QY:^UI\$QD:^UI\$
QT:^UI\$QR:^UI\$QR:^UB\$QL:^UB\$QY:^UB\$QI:^UB\$QT:^UB\$GI-5CGUGS-5CGB10CGQ0JT\$\$
----- END TECO SIGNATURE BLOCK -----
(Don't forget: ^ in TECO means just that, and \$ means press the Esc key!)

Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>
To: Pete Maclean <pete@maclean.com>

13 July 2019 at 15:36

Dear Mr Maclean:

Sorry for not replying sooner—I was under the misapprehension that the RAR file contained the licence code. Yes, that was my intention all along. I'd love to see Eudora compiled with the 2004 libraries.

The licence code is as follows: 726AD67CC3013C0CE4 and the validation key is 3E979-013-8D482299DE

You will need both to install the software, though quite how that works I forget. In any case, there you have it. I'll try to install it on my own computer, as well.

Regards,

Nicholas E. Werner-Matavka

>

On Mon, 24 Jun 2019 at 15:07, Pete Maclean <pete@maclean.com> wrote:

I am curious to try a build of the main Eudora executable with this Stingray 2004 package. Would you like me to do so? If yes then I need a Validation Code to install Stingray.

Pete Maclean

At 07:38 PM 6/20/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do. They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process. I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out.Â Stingray 2004 is available here:Â https://drive.google.com/open?id=1KHZMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother.Â I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed.Â Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive.Â Let's save that for if we really must.

Cordially,

N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean <pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical.Â Which, as you say, means using the Stingray library, no matter how distasteful we find that to be.Â And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it.Â Even if that money were suddenly dropped in our laps, we might still have two significant obstacles.Â One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used.Â The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know.Â In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy.Â I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release.Â Specifically this is "Microsoft Visual Studio .NET 2003".Â I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables.Â The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise.Â Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking. This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,

N. E. Werner-Matavka

--
----- BEGIN TECO SIGNATURE BLOCK -----
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC\$
QA:^US\$QP:^US\$QD:^US\$QI:^US\$QA:^US\$QM:^UQ\$QG:^UQ\$QA:^UQ\$QP:^UQ\$
QE:^UQ\$QO:^UU\$QC:^UU\$QH:^UU\$QI:^UU\$QD:^UU\$QM:^UI\$QY:^UI\$QD:^UI\$
QT:^UI\$QR:^UI\$QR:^UB\$QL:^UB\$QY:^UB\$QI:^UB\$QT:^UB\$GI-5CGUGS-5CGB10CGQ0JT\$\$
----- END TECO SIGNATURE BLOCK -----
(Don't forget: ^ in TECO means just that, and \$ means press the Esc key!)

Pete Maclean <pete@maclean.com>

14 July 2019 at 14:24

To: Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

Thank you. I now have it installed. And will start working with it tomorrow.

Pete Maclean

At 03:36 PM 7/13/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

Sorry for not replying sooner—I was under the misapprehension that the RAR file contained the licence code. Yes, that was my intention all along. I'd love to see Eudora compiled with the 2004 libraries.

The licence code is as follows: 726AD67CC3013C0CE4 and the validation key is 3E979-013-8D482299DE

You will need both to install the software, though quite how that works I forget. In any case, there you have it. I'll try to install it on my own computer, as well.

Regards,

Nicholas E. Werner-Matavka

>

On Mon, 24 Jun 2019 at 15:07, Pete Maclean <pete@maclean.com> wrote:

I am curious to try a build of the main Eudora executable with this Stingray 2004 package. Would you like me to do so? If yes then I need a Validation Code to install Stingray.

Pete Maclean

At 07:38 PM 6/20/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do. They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process. I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out. Stingray 2004 is available here: https://drive.google.com/open?id=1KHZMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother. I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed. Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive. Let's save that for if we really must.

Cordially,

N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean <pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical. Which, as you say, means using the Stingray library, no matter how distasteful we find that to be. And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it. Even if that money were suddenly dropped in our laps, we might still have two significant obstacles. One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used. The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know. In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy. I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release. Specifically this is "Microsoft Visual Studio .NET 2003". I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables. The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise.Ã,Ã Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now. That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices.Ã,Ã Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking.Ã,Ã This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing.Ã,Ã I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,
N. E. Werner-Matavka

```
--  
----- BEGIN TECO SIGNATURE BLOCK -----  
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC$  
QA:^US$QP:^US$QD:^US$QI:^US$QA:^US$QM:^UQ$QG:^UQ$QA:^UQ$QP:^UQ$  
QE:^UQ$QO:^UU$QC:^UU$QH:^UU$QI:^UU$QD:^UU$QM:^UI$QY:^UI$QD:^UI$  
QT:^UI$QR:^UI$QR:^UB$QL:^UB$QY:^UB$QI:^UB$QT:^UB$GI-5CGUGS-5CGB10CGQ0JT$$  
----- END TECO SIGNATURE BLOCK -----  
(Don't forget: ^ in TECO means just that, and $ means press the Esc key!)
```

Pete Maclean <pete@maclean.com>
To: Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

19 July 2019 at 15:12

I have some significant progress to report. I now have a clean build of the Qualcomm version of the Stingray library and a clean compilation of the main Eudora/Hermes executable.

There are some caveats: First, I have made all customizations to Stingray for which there are clues in the Eudora sources but I cannot be sure that I have made them fully and correctly. In fact, in one case I am sure I have done it only partially but it will only be once I see the program working that I will be easily able to fix it. Second, getting a clean compilation of the Eudora executable took a lot of work and while I elegantly fixed a few things there are other things that I fixed by brute-force means that will have to be cleaned up later. Third, I think that getting a clean compilation of the executable is a major step but getting a clean link looks like it may take a lot more work.

Pete Maclean

At 03:36 PM 7/13/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

Sorry for not replying sooner—I was under the misapprehension that the RAR file contained the licence code. Yes, that was my intention all along. I'd love to see Eudora compiled with the 2004 libraries.

The licence code is as follows: 726AD67CC3013C0CE4 and the validation key is 3E979-013-8D482299DE

You will need both to install the software, though quite how that works I forget. In any case, there you have it. I'll try to install it on my own computer, as well.

Regards,

Nicholas E. Werner-Matavka

>

On Mon, 24 Jun 2019 at 15:07, Pete Maclean <pete@maclean.com> wrote:

I am curious to try a build of the main Eudora executable with this Stingray 2004 package. Would you like me to do so? If yes then I need a Validation Code to install Stingray.

Pete Maclean

At 07:38 PM 6/20/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do. They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process. I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out. Stingray 2004 is available here: https://drive.google.com/open?id=1KHzMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother. I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed. Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive. Let's save that for if we really must.

Cordially,

N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean <pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical. Which, as you say, means using the Stingray library, no matter how distasteful we find that to be. And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it. Even if that money were suddenly dropped in our laps, we might still have two significant obstacles. One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used. The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know. In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy. I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release. Specifically this is "Microsoft Visual Studio .NET 2003". I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables. The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics. In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you.

It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices. Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking. This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team. With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate

the alternative.

Please inform me in due time of which path you feel most comfortable pursuing. I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,
N. E. Werner-Matavka

--

----- BEGIN TECO SIGNATURE BLOCK -----

```
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC$
QA:^US$QP:^US$QD:^US$QI:^US$QA:^US$QM:^UQ$QG:^UQ$QA:^UQ$QP:^UQ$
QE:^UQ$QO:^UU$QC:^UU$QH:^UU$QI:^UU$QD:^UU$QM:^UI$QY:^UI$QD:^UI$
QT:^UI$QR:^UI$QR:^UB$QL:^UB$QY:^UB$QI:^UB$QT:^UB$GI-5CGUGS-5CGB10CGQ0JT$$
```

----- END TECO SIGNATURE BLOCK -----

(Don't forget: ^ in TECO means just that, and \$ means press the Esc key!)

Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>
To: Pete Maclean <pete@maclean.com>

20 July 2019 at 09:40

Dear Mr Maclean:

I'm not, as a rule, given to beating about the bush, and this time is no exception; this is, without qualification, excellent news. Having a clean build means we are well on the way to being able to remedy the issues that matter. In particular, I'd like you to have a look at the Chromium Embedded Framework (located at <https://bitbucket.org/chromiumembedded/cef/src>) and see how best to replace the links to MSHTML.DLL. As long as eudora.exe is still linked to aforementioned file, we are essentially staring down a gargantuan security hole, and W3C standards compliance? What standards?

We'll also need to work out how spell check fits into this mess. The best spell-check module, and I have done my due diligence on this matter, is Hunspell, which also happens to support a lot of languages. I would have recommended Nuspell, which is written in pure C++, but given its unfinished nature at the present time, I can not do so.

The Stingray customisations take top priority, of course; be sure to document your work as you go, so that we have some form of turnover guide (What if you disappeared to-morrow?)

The two deal-breakers, at least for ex-Eudora users, are SSL and language support. You fixed the SSL (thanks and congratulations are in order), UTF-8 still left to do, and then we're also missing stuff like Sentry. Problem is, that won't do much to attract new users, of which we need many. Public relations WILL attract new users, and I'm fairly good at that... but the other cheek of that particular arse, so to speak, is features.

Once we have some form of Eudora compiled, linked, re-named HERMES Mail, and packaged up nicely into a Windows executable, we can discuss feature additions. What I have in mind are mostly quality-of-life improvements, things that would take max. 100 lines of code each, but that would make us tremendously competitive with even something like Postbox.

What we need to discuss is essentially workflow. Do you want to concentrate on the work necessary to compile and link Eudora in its present state, or make the necessary changes and THEN try to link it? Either is fine, in my opinion; you're the programmer.

Cordially,

Nicholas Edward Werner-Matavka

On Fri, 19 Jul 2019 at 15:12, Pete Maclean <pete@maclean.com> wrote:

I have some significant progress to report. I now have a clean build of the Qualcomm version of the Stingray library and a clean compilation of the main Eudora/Hermes executable.

There are some caveats: First, I have made all customizations to Stingray for which there are

clues in the Eudora sources but I cannot be sure that I have made them fully and correctly. In fact, in one case I am sure I have done it only partially but it will only be once I see the program working that I will be easily able to fix it. Second, getting a clean compilation of the Eudora executable took a lot of work and while I elegantly fixed a few things there are other things that I fixed by brute-force means that will have to be cleaned up later. Third, I think that getting a clean compilation of the executable is a major step but getting a clean link looks like it may take a lot more work.

Pete Maclean

At 03:36 PM 7/13/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

Sorry for not replying sooner—I was under the misapprehension that the RAR file contained the licence code. Yes, that was my intention all along. I'd love to see Eudora compiled with the 2004 libraries.

The licence code is as follows: 726AD67CC3013C0CE4 and the validation key is 3E979-013-8D482299DE

You will need both to install the software, though quite how that works I forget. In any case, there you have it. I'll try to install it on my own computer, as well.

Regards,

Nicholas E. Werner-Matavka

>

On Mon, 24 Jun 2019 at 15:07, Pete Maclean <pete@maclean.com> wrote:

I am curious to try a build of the main Eudora executable with this Stingray 2004 package. Would you like me to do so? If yes then I need a Validation Code to install Stingray.

Pete Maclean

At 07:38 PM 6/20/2019, Nick Werner-Matavka wrote:

Dear Mr Maclean:

While I view the business practices of Quovadx (the current publishers of Objective Toolkit/"Stingray Desktop") repugnant in the extreme, this has to do with their point-blank refusal to provide a recent version of the libraries free of charge for evaluation purposes, as Stingray Inc. used to do. They did, however, grant us a limited-use licence for Stingray 2004; ergo, financial considerations only played a minor role in my thought process. I still feel the need to transition from this old version to either the modern Stingray Desktop (which, as you know, will be expensive) or to something free of cost---but given our issues in this area, I've mentally downgraded this (monumental) task in priority.

The only hurdle, at the moment, is the fact that Qualcomm made certain undisclosed modifications to OT/Stingray, but I feel confident we'll be able to work something out. Stingray

2004 is available here: https://drive.google.com/open?id=1KHzMSwJG2dfmPtdMG25HWQCv4KTe3G_C

As for using VS .NET 2003: don't bother. I decided to distribute the QCSSL package in an InstallShield that will ensure the redistributable package is installed. Furthermore, I think using an out-of-date version of something that we have a recent version of, is counter-productive. Let's save that for if we really must.

Cordially,
N. E. W. M.

On Mon, 17 Jun 2019 at 12:15, Pete Maclean
<pete@maclean.com> wrote:

Dear Mr. Werner-Matavka,

It remains my contention that the only sensible strategy for us, given our talents, circumstances and limitations, is to start by building Hermes with code that is as close to the original Eudora code as practical. Which, as you say, means using the Stingray library, no matter how distasteful we find that to be. And that in turn presumably means coming up with something in the neighbourhood of either \$2,800 or \$3,600 to obtain a license for it. Even if that money were suddenly dropped in our laps, we might still have two significant obstacles. One is that Rogue Wave may no longer have the exact version of the toolkit that the Eudora creators used. The other is that, if I understand correctly, the Eudora creators made modifications to that library, modifications that we have no documentation on as far as I know. In principle I am willing to put up a sizeable chunk of the license fee out of my own pocket but only if I am very satisfied that we have enough expertise committed to carry the project through.

Recently I have made one small step that will facilitate this strategy. I have found and installed a version of Microsoft Visual Studio that is very close, although maybe not identical, to the version used to build the last Eudora release. Specifically this is "Microsoft Visual Studio .NET 2003". I am using this already to create the next release of Hermes QCSSL which will give it the advantage of using the same C/C++ runtime that Eudora 7.1 uses, thus avoiding the need for users to install the Visual C 2015 redistributables. The other advantage of using this version of Visual Studio is that we can compile the Eudora code without having to deal with the 1,000-plus errors that the 2015 compiler finds to complain about.

Pete Maclean

At 10:07 AM 6/12/2019, Nicholas Edward Werner-Matavka wrote:

Gentlemen:

I have been thinking long and hard about the task we are faced with, the recurrent schedule slippage, and my personal principles of business ethics.

In retrospect, it's become abundantly clear to me that Mr MacLean was right, and I was wrong, qua which toolkit to prioritise. Had we gone with the existing Stingray-based code, the unadorned fact is that we would have a fully functional successor to Eudora by now.

That said, it's also clear that much work has been done on the MFC-based port, that Mr Thygesen is a specialist in this field, and that it's what the Qualcomm team would have wanted.

I can not, with any degree of honesty, say that it was my fault, because no human being could have anticipated the insurmountable complexity or

time demand for the job---but I'll gladly admit that the facts before us now are not in my favour, and I'll therefore leave the decision up to you. It was, and it still remains, my opinion that truly Open Source software can not ethically be built with expensive, proprietary tools from vendors that employ unconscionable business practices.Ã,Ã Unfortunately, in a collision of sound principles and ugly reality, the principles are almost never the victors.

I'm sure that I don't need to explain what still needs done, if we make the collective decision to prioritise the Stingray codebase, but in brief, we would definitely need to adapt both Hunspell and the Chromium Embedded Toolkit to fill the shoes of their proprietary counterparts, and Unicode support remains sorely lacking.Ã,Ã This is most definitely not an "effortless" alternative, especially because the version of Stingray we are licenced to use may, or may not, be the one relied upon by the Qualcomm team.

With the general mood of distrust and unfounded belief that our work proceeds "behind closed doors", however, I feel honour-bound to reiterate the alternative.

Please inform me in due time of which path you feel most comfortable pursuing.Ã,Ã I'm sure you know what path I would prefer, but, again, this is entirely open to discussion.

Yours most sincerely,
N. E. Werner-Matavka

--

----- BEGIN TECO SIGNATURE BLOCK -----

```
32UD44UE97UR99UM101UA104UT106UO107UG110UL111UY114UP115UH116UI117UC$
QA:^US$QP:^US$QD:^US$QI:^US$QA:^US$QM:^UQ$QG:^UQ$QA:^UQ$QP:^UQ$
QE:^UQ$QO:^UU$QC:^UU$QH:^UU$QI:^UU$QD:^UU$QM:^UI$QY:^UI$QD:^UI$
QT:^UI$QR:^UI$QR:^UB$QL:^UB$QY:^UB$QI:^UB$QT:^UB$GI-5CGUGS-5CGB10CGQ0JT$$
```

----- END TECO SIGNATURE BLOCK -----

(Don't forget: ^ in TECO means just that, and \$ means press the Esc key!)

Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

27 July 2019 at 19:53

To: Eudora for Windows <eudora-win@hades.listmoms.net>, Eudora-Dev <eudora-dev@hades.listmoms.net>

Hopefully this puts lead in everyone's pencil...

----- Forwarded message -----

From: Pete Maclean <pete@maclean.com>

Date: Fri, 19 Jul 2019 at 15:12

Subject: Re: State of HERMES (project)

To: Nick Werner-Matavka <n.theodore.matavka.files@gmail.com>

I have some significant progress to report. I now have a clean build of the Qualcomm version of the Stingray library and a clean compilation of the main Eudora/Hermes executable.

There are some caveats: First, I have made all customizations to Stingray for which there are clues in the Eudora sources but I cannot be sure that I have made them fully and correctly. In fact, in one case I am sure I have done it only partially but it will only be once I see the program working that I will be easily able to fix it. Second, getting a clean compilation of the Eudora executable took a lot of work and while I elegantly fixed a few things there are other things that I fixed by brute-force means that will have to be cleaned up later. Third, I think that getting a clean compilation of the executable is a major step but getting a clean link looks like it may take a lot more work.

