



I E T F[®]

IETF ADMINISTRATION LLC

Questions & Answers: RFC Errata Merge Tool

17 April 2019

- 1) In bullet 8 of the RFP, you say that bidders for "the Datatracker Meeting Application Improvements" should indicate if they want to be considered as IDIQ providers. Presumably, you meant to refer to this project? If a bidder does want to be considered, is there any additional information that should be provided?

That was a copy-and-paste error in the RFP. Please express interest your proposal for this project if you want to be considered for the IDIQ as well as the RFC Errata Merge Tool. We will ask for more information if we proceed with setting up an IDIQ agreement with you.

There is more information available about the IDIQ here:
<https://www.ietf.org/documents/252/rfp-software-idiq-2019-01-23.pdf>

The intent is that a response to this RFP would be used as the example project rather than the hypothetical rfcdiff described in that document.

- 2) Is there a particular programming language or framework that you would prefer we use to develop the tool (e.g. Python Django)?

Python is highly preferred. This project does not need to use Django.

- 3) Is there a particular operating system or web server on which you expect the server side of this application to be supported (e.g. an Apache Web Server running on Debian)?

At the moment, it would be Apache on OpenSuse, but the development should be portable to any modern Linux system. It is desirable that the code not be Apache specific; allowing deployment with nginx, for example, would be desirable.

- 4) Is there any sort of access control or authentication required for the use of the tool, or will it be anonymously open to everyone?

No authentication will be required at this time. A future extension might authenticate against IETF Datatracker credentials to allow Area Directors see proposed errata inline, but there is no timeline for such an extension.

- 5) There are several instances of blue underlined text in the SOW

that appear to be links, but they are not. Could you please provide links to all the relevant documents, including: database of errata, a simplistic prototype, the Rfcdiff tool, and the Statcounter.

Please refetch the RFP from

https://www.ietf.org/documents/277/RFC_Errata_Merge_Tool_RFP_04-08-19_Appended.pdf.

The links have been added as an appendix.

6) Will we receive source code for the “simplistic prototype”? If so, could you provide access now, so that we can refer to it when putting together our bid?

It is open source. It is located at one of the links above:

<https://github.com/adamroach/patch-errata>

Also, there are two example outputs that may help:

<https://adamroach.github.io/patched-rfcs/rfc/rfc3261.html>

<https://adamroach.github.io/patched-rfcs/rfc/rfc5246.html>

7) In the section "Locating Errata Placement" bullet #3, would it be permissible to always ignore headers and footers when searching an RFC for matching, with the ability to strip things that look like headers and footers on a failed match?

Any algorithm that yields the same result as the steps in the SOW is allowable.

8) The next to last paragraph in the section "Locating Errata Placement" appears to be contradictory assuming the if statements are intended to be read in sequence. Specifically if the match cannot be found in a section then placing the errata at the section level would not be possible.

The point here is to use the section number information from the reported errata, not what you matched (or failed to match) in the document.

9) For placement of inline errata, is the supplementary information to be placed at the end of the paragraph or immediately following the line where the correction is placed?

Please see the prototype. However, the prototype is a proof of concept, not a replacement for the requirements in the SOW. The prototype is not a reference implementation.

10) If multiple inline errata are made to the same location, is it acceptable to have a single expand/collapse control for all errata in the same line or paragraph?

Yes, as long as the errata can be distinguished, and it is clear what happened to get the resulting text.

11) For errata that are placed at the section level, are they to be supplied with a collapse feature?

Yes, but they should start expanded.

12) While the data is currently not being defined as displayed, the errata.json file that I down loaded has several occurrences of the verifier name of null and multiple different names being provided for id 99.

This isn't a question, but we will consider it a bug report.

This is an example for the text that says: "reasonable requests for changes will be accommodated".

13) Should there be a difference in formatting for technical and editorial errata? Specifically, should either the styles be different or differences in what errata are inserted based on command line options?

We are not asking for different formatting at this time, but it would be good to make sure it is not hard to add in the future.

14) The section field in the JSON appears to be emitted as a number in many cases rather than as a string. This means that you get "section":9.3 rather than "section":"9.3" which is probably not desirable.

This isn't a question, but we will consider it a bug report.

This is an example for the text that says: "reasonable requests for changes will be accommodated".

15) Is there an intention that a pass over the errata database is going to be made to clean up issues discovered during this project to allow for inlining to occur better or should the tool become highly heuristic about identifying locations?

The provided tool should be heuristic, but not to the point of encoding information about specific RFCs or specific errata.

A comprehensive pass over the errata database as part of this project is not anticipated, but problems should be reported, and if they are sufficiently severe, we can work with the RFC Editor to prioritize fixing them.

16) Is it reasonable to assume that the server which is driving this tool will also be on US Pacific Time as well so that issues dealing with when the transition from PST to PDT is made can be ignored?

No, Please do not make that assumption; however, the errors that it introduces will not likely be important. Variations by up to an hour due to daylight savings time changes are permissible.

17) There is a statement that no inline styling is to be placed in the HTML file, this means that if one is doing a copy-and-paste operation from the file to a different location such as an email program will lose formatting (such as colors). Is this really an acceptable outcome? It used to be that browsers did this inlining automatically, but they no longer do so. This is presumably to better support alternative style sheets, but it does mean that the information is not presently available for copy-and-paste operations.

Yes, this is very much what is being asked for.

18) On the assumption that python is going to be the development tool, what version(s) of python are required to be supported?

Python 3. Note that Python 2 is planned for end-of-life at the end of this year. Ideally, the code will work with Python 3.5 up to Python 3.7.

19) Testing level: What level of verification of integration is going to be required by the contract? Is spot checking of items sufficient or is an exhaustive of a certain set or percentage of errata sufficient? The assumption is that this would be beyond the basic testing for regression and designed test cases.

Expect a selected set of RFCs to be used for determining when the project is complete. The set of RFCs has not been selected yet.

20) Are there any implications of the move from V2 to V3 xml2rfc vocabulary that I need to be aware of while implementing this?

No. This project works only on text files, and the primary intention is to work on text files published to date. Working on text files generated by the v3 formatters is anticipated in the future, but it is not required by this project.

21) Is there going to be a press to do the rest of the rfc2html code on this at some point in the future? Does this need to be allowed for?

We do not envision that it would be applied to the result of this project incrementally. Rather, new project would work with the v3 HTML renderer in the future.

22) How should the interactions of multiple errata be handled? There are a number of different ways that multiple errata can interact and depending on the choices made the amount of work is highly variable. Which of the following cases are of importance?

- * Newer errata is nested in older errata using original text;
- * Newer errata is nested in older errata using updated text;
- * Older errata is nested in newer errata using original text;
- * Older errata is nested in newer errata using updated text; or
- * Newer errata intersects with older errata.

Does it matter if the newer errata is reflected in the updated text as opposed to being a "standalone" errata?

If it is very straightforward to inline a particular nesting of errata, that would probably best serve the reader. However, listing the errata at the section level is acceptable when nesting occurs.

23) Line wrapping is a mess and is not well reflected in the example presented in the RFP. There are number of different possible approaches.

- * Simple delete does not have any problems with long lines because it just colors the original text. It also makes the different cases simpler as the coloring can just be expanded with any issues.
- * Simple replacement is easy as no line wrapping is needed. However it will make the printing potentially poor as the page may be sized to fit the longer line.
- * Simple replace + wrap: This is easy as no wrapping is needed. Some amount of wrapping will be done by the HTML as long lines are permitted to be automatically wrapped. This should make printing look nicer.
- * Insert break: This is how the example in the RFP was done, but as you can see it does not always make sense and destroys the flow of text as paragraphs are murdered.
- * Rewrap inserted line: This basically does the same line wrapping that is done by xml2rfc, but is harder because of the spans, and could be done at a longer line length to deal with some of the issues that the example identifies. This requires that the text can identify what is a paragraph and what is examples in order to wrap correctly. As only a single line at a time is wrapped it may be example to wrap lines that are just too long.
- * Rewrap paragraph: This is the hardest to implement as it requires both dealing with the inserts of hidden items while

wrapping as well as needing to correctly identify what is a paragraph as opposed to just running text of examples or figures which just happen to have long lines.

Please keep the implementation as simple as possible. Once the IETF community has some experience with the Errata Merge Tool, this is a place where incremental improvements can be made in the future.