

Overview

The RFC Editor plans to produce a canonical RFC document in XML using the xml2rfc v3 grammar, and to publish the RFC in several Publication Formats as defined in RFC6949.

This project will result in one or more applications to produce the Text, HTML, and PDF publication formats from an xml2rfc v3 source document. It will also produce a mechanism to create an xml2rfc v3 document from an xml2rfc v2 document, and a mechanism to conditioning an xml2rfc v3 document entry into the RFC archives.

Deliverables/Tasks

- Design the command line interface(s)
- Develop the conversion application(s)
- Demonstrate the conversion of a specified set of test document in a standalone environment
- Demonstrate the conversion of a specified set of test documents using a web service
- Provide an extensible test suite for the application(s)
- Provide documentation, and training for the RFC Production Center staff

Detailed Description and Requirements

Each application must run standalone in Linux, os/x, and Windows environments. The application will also be made available as part of a web-service offering the translation, similar to what's available now at <http://xml2rfc.tools.ietf.org/>.

The developer will work with the Program Manager to agree on how each application is invoked before beginning development work.

The project will use the following document(s) as the primary development and test targets:

`<https://github.com/hlflanagan/draft-hildebrand-html-rfc/blob/master/test.3.xml>`

`<http://trac.tools.ietf.org/tools/xml2rfc/trac/browser/trunk/cli/tests/input/draft-miek-test.xml>`

`<http://trac.tools.ietf.org/tools/xml2rfc/trac/browser/trunk/cli/tests/input/draft-miek-test-v3.xml>`

`<http://datatracker.ietf.org/doc/draft-hoffman-rfcexamples/>`

The application suite will accept input using either the xml2rfc v3 grammar (as described in draft-hoffman-xml2rfc) or in the xml2rfc v2 grammar (as described in draft-reschke-

xml2rfc). When processing an xml2rfc v2 document, the suite will produce a version converted into the xml2rfc v3 grammar. We expect the implementation of that conversion to be informed by the substantially complete grammar converter developed by the volunteer community at <<http://zinfandel.tools.ietf.org/~tonyh/convertv2v3/convertv2v3.txt>>).

The application(s) will produce output as specified in the following documents:

Format	Specifications
Text	draft-flanagan-plaintext draft-iab-styleguide (section 4)
HTML	draft-hildebrand-html-rfc
PDF	draft-hansen-rfc-use-of-pdf

It is acceptable (as xml2rfc does now) to produce variations on those output formats, and other output formats, controlled by the application invocation or instructions in the document source.

There is a hand-crafted example of intended html output at <https://github.com/hildjj/draft-hildebrand-html-rfc/blob/master/test.3.html>. Where it conflicts with the definitions in draft-hildebrand-html-rfc, the latter, defining, document takes precedence.

The application(s) will correctly process any provided SVG (as defined by draft-brownlee-svg-rfc, according to the each output specification.

When producing an xml2rfc v3 output document the application will translate any deprecated elements in the v3 vocabulary using the constructs described in draft-hoffman-xml2rfc. This processing will occur both when the input document is v2 and when it is v3.

When creating an xml2rfc v3 document conditioned for entry into the RFC archives, the application will perform the following additional processing:

- The appropriate boilerplate detailed in RFC5741 will be placed in the document's <boilerplate> element. Any content within the <boilerplate> element from the input document will be replaced. The application will generate a warning if the replaced text was not identical to the generated text. (Note that other applications, such as the id submission tool or the idnits tool will likely treat this condition as an error). The application will use the ipr, category, submission, and consensus attributes of the <rfc> element to determine which RFC5741 boilerplate to include. See Appendix A of draft-hoffman-xml2rfc.
- The following attributes will be generated, replacing any attribute values that might appear in the input document. The application will generate a warning if the replaced values are not identical to the generated values.

- The pn attribute of the section, abstract, note, figure, table, artwork, aside, blockquote, dt, li, sourcecode, and t elements will be calculated and formatted as described in draft-hoffman-xml2rfc Appendix B.2.1
- The slugifiedName attribute of the name element will be calculated and formatted as described in that same appendix.
- Any xincludes, include-like processing instructions, src= references in <code> and <artwork> elements, and any external entity definitions will be expanded into the output document. Since this expansion is part of normal xml parsing, the application is expected to utilize its underlying xml parser to meet this requirement.
- Any comments, processing instructions, or DTD references still remaining in the document after the above processing will be removed
- The output document will be produced with a 'pretty-printed' whitespace and line-break structure.

When presented with an input document that contains errors, the diagnostic output of the application(s) must help an author or the RFC Editor quickly locate and correct the source of each error. When possible, the application(s) should suggest corrective actions.

The requirements on the formats defined above are expected to change over time. The application must be easy to modify and maintain. An application written in a language already familiar to most of the volunteer support community is preferred.