

RFC Format BoF

IETF 88

Vancouver, BC, Canada

Homework

Have you read the following:

<https://www.rfc-editor.org/rse/wiki/doku.php?id=design:start>

This is a public read-only wiki space

Agenda

- Background
- The RFC Format Design Team
- Current Status
- Next Steps
- Expected questions

Background

The format announcement in May indicated several things:

- the canonical format we are exploring for RFCs is XML
- four publication formats will be created from that XML: HTML, EPUB, text and PDF
- non-ASCII characters would be allowed in a controlled fashion

<http://www.rfc-editor.org/pipermail/rfc-interest/2013-May/005584.html>

RFC Format Design Team

- An RFC format design team was put together during IETF 87 in Berlin to clear up the details implied by those statements

<https://www.rfc-editor.org/rse/wiki/doku.php?id=design:design-team>

Many thanks to Nevil Brownlee (ISE), Tony Hansen, Joe Hildebrand, Paul Hoffman, Ted Lemon, Julian Reschke, Adam Roach, Alice Russo, Robert Sparks (Tools Team liaison), and Dave Thaler for their active participation

Current Status (1)

- In Progress: documenting the current vocabulary and description of the current xml2rfc DTD and bring it up to date and drafting the proposed changes going forward
 - <http://tools.ietf.org/html/draft-reschke-xml2rfc>
- In Progress: requirements for the HTML and text formats
 - EPUB should be visually derived from HTML, and PDF from text (plus images)
 - acceptance of draft-hildebrand-html-rfc as a solid starting place for the HTML details
 - <http://cursive.net/draft-hildebrand-html-rfc.html>

Current Status (2)

- Agreement in principle to include non-ASCII characters in RFCs
 - details being worked out in conjunction with the i18n program of the IAB
- A high level work flow for how the tool will be used in production by authors and the RFC Editor
<https://www.rfc-editor.org/rse/wiki/doku.php?id=design:producing-output>
- In progress: details around the use of images
 - RFCs will be able to have embedded SVG art for figures, at the discretion of the authors<https://www.rfc-editor.org/rse/wiki/doku.php?id=design:image-requirements>

Next Steps

- Finish the xml2rfc v2 and v3 descriptions and requirements
- Finish draft-hildebrand-html-rfc
- Create the SoW to start on the specs, followed by the development phase
- Discuss what, if any, changes should be phased in versus a formal cut-over
- Discuss how this affects the submission process and I-D format

Expected Question #1

- What about things like look-and-feel?
 - across the board, images and tables will be restricted to no more than 80 characters
 - for HTML and EPUB, we are expecting reflowable text, which will change the look of an RFC viewed through the rendering agents for those protocols
 - for HTML, HTML-savvy people will be able to control how an RFC looks, the fonts used, size of fonts, layout of headers
 - The RFC Editor will have a layout they publish

Expected Question #2

- How will non-ASCII characters be handled?
 - Consensus reached in discussions with the i18n; working on clarifying the text
 - Non-ASCII should be consistent across all publication formats (text, PDF, HTML, and EPUB).

Non-ASCII examples

(color and boldface highlight examples – their use is not part of the proposal for non-ASCII text)

- CURRENT (draft-ietf-precis-framework) :
However, the problem is made more serious by introducing the full range of Unicode code points into protocol strings. For example, the characters U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 from the Cherokee block look similar to the ASCII characters "STPETER" as they might look when presented using a "creative" font family.
- PROPOSED/NEW:
However, the problem is made more serious by introducing the full range of Unicode code points into protocol strings. For example, the characters U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 (**STPETER**) from the Cherokee block look similar to the ASCII characters "STPETER" as they might look when presented using a "creative" font family.
- ALSO ACCEPTABLE:
However, the problem is made more serious by introducing the full range of Unicode code points into protocol strings. For example, the characters **"STPETER" (U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2)** from the Cherokee block look similar to the ASCII characters "STPETER" as they might look when presented using a "creative" font family.

Body Text – example

(color and boldface highlight examples – their use is not part of the proposal for non-ASCII text)

- Current (RFC 6912)
- The IAB recommendations do, however, leave some issues open that need to be addressed. It is not clear that all code points permitted under IDNA2008 that have a General_Category of Lo or Lm are appropriate for a zone such as the root zone. To take but one example, the code point U+02BC (MODIFIER LETTER APOSTROPHE) has a General_Category of Lm. In practically every rendering (and we are unaware of an exception), U+02BC is indistinguishable from U+2019 (RIGHT SINGLE QUOTATION MARK), which has a General_Category of Pf (Final_Punctuation). U+02BC will also be read by large numbers of people as being the same character as U+0027 (APOSTROPHE), which has a General_Category of Po (Other_Punctuation), and some computer systems may treat U+02BC as U+0027. U+02BC is PROTOCOL VALID (PVALID) under IDNA2008 (see the IDNA Code Points document [RFC5892]), whereas both other code points are DISALLOWED. So, to begin with, it is plain that not every code point with a General_Category of Ll, Lo, or Lm is consistent with the type of conservatism principle discussed in Section 4.1 below or the previous IAB recommendations.

- Proposed/New(RFC 6912)

The IAB recommendations do, however, leave some issues open that need to be addressed. It is not clear that all code points permitted under IDNA2008 that have a General_Category of Lo or Lm are appropriate for a zone such as the root zone. To take but one example, the code point **U+02BC (' , MODIFIER LETTER APOSTROPHE)** has a General_Category of Lm. In practically every rendering (and we are unaware of an exception), U+02BC is indistinguishable from **U+2019 (' , RIGHT SINGLE QUOTATION MARK)**, which has a General_Category of Pf (Final Punctuation). U+02BC will also be read by large numbers of people as being the same character as **U+0027 (' , APOSTROPHE)**, which has a General_Category of Po (Other Punctuation), and some computer systems may treat U+02BC as U+0027. U+02BC is PROTOCOL VALID (PVALID) under IDNA2008 (see the IDNA Code Points document [RFC5892]), whereas both other code points are DISALLOWED. So, to begin with, it is plain that not every code point with a General_Category of Ll, Lo, or Lm is consistent with the type of conservatism principle discussed in Section 4.1 below or the previous IAB recommendations.