Network Working Group
Internet-Draft
Intended status: Informational
Expires: April 4, 2019

H. Levkowetz
Elf Tools AB

October 1, 2018

Implementation notes for RFC7991, "The 'xml2rfc' Version 3 Vocabulary"
draft-levkowetz-xml2rfc-v3-implementation-notes-03

Abstract

This memo documents issues and observations found while implementing RFC 7991. Individual notes are organised into separate sections, depending on their character.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

Implementation of tool support for [RFC7991] and related specifications has been done during 2017 and 2018, split in the following individual parts, all implemented as individual modes of the python-based xml2rfc processor [XML2RFC]:

- An XML converter from vocabulary version 2 [RFC7749] to version 3 [RFC7991]

- A Normalisation processor, "PrepTool", [RFC7997]

- An XML to plain text converter [RFC7994] for the version 3 vocabulary

- An XML to html converter [RFC7992] for the version 3 vocabulary (work in progress as of 28 Sep. 2018)

- A HTML to PDF converter [RFC7995] for the version 3 vocabulary (pending as of 28 Sep. 2018)

During the implementation work, a number of issues with the specification has been found (this was expected at the outset by all parties) and a number of observations has been made about limitations of the specification and vocabulary version 3 schema, and also limitations in the specification of the work to be done.

The purpose of this memo is to collect those issues and observations in one place.

When this memo says 'the current version of xml2rfc', it refers to the latest release of the xml2rfc processor available from the PyPi package repository [1] at the date this document was published, as given above. As of 28 Sep. 2018, this was version 2.10.3.

2. Fitness for Purpose

The introduction to [RFC7991] states:

> "This document defines the "xml2rfc" version 3 vocabulary: an XML-based language used for writing RFCs and Internet-Drafts. It is heavily derived from the version 2 vocabulary that is also under discussion. This document obsoletes the v2 grammar described in RFC 7749."

However, an unstated assumption seems to have been that the new tools and formatters would be used primarily to produce HTML output, in order to transition to publication of renderings of RFCs in more modern formats than plain-text ASCII.

This is a reasonable and worthwhile goal, but as a result, the schema as specified in [RFC7991] has some drawbacks compared with the version 2 vocabulary when used to produce Internet-Drafts in the text format common within the IETF (Internet Engineering Task Force) at this time.

2.1. Degraded Table of Contents

Lack of pagination has little impact on direct online readability, but when comparing the output of the new text formatter with the old one, one aspect leaps out: Since there is no pagination, the table of contents simply lists the section headers to a certain depth, without any accompanying page numbers. This makes a surprising difference in how useful the table of contents is in getting an initial feel for the document. The at-a-glance information which lets a reader know if this is a document of 10 pages or 100 is simply lacking.

Recommendation: Add support for pagination in a future version of the text formatter.

Implementation: None in the current version of xml2rfc.

2.2. RFC Publication Date Policy

The specification [RFC7998] says that an error should be generated if a <date> specification is found with missing elements; but the RFC Editor publishes documents (except for April 1st RFCs) with only year and month, no day of month. The specification disallows this, and in effect makes it impossible for the RFC Editor to publish documents according to the current policy regarding publication date format.

Recommendation: Revert to to the old behaviour, where the tool in RFC mode would issue a date with or without day depending on whether the <date> element had a day attribute or not.

Implementation: All elements of <date> are required in the current version of xml2rfc.

3. Schema Issues

3.1. RFC 7991

3.1.1. In Section 2.5.5, "name" Attribute

> "A filename suitable for the contents (such as for extraction to a local file)."

Given the existing use of "name" on <seriesInfo>, this attribute name has a semantic dissonance.

Recommendation: Deprecate "name" for use on <artwork> and <sourcecode>, and instead use "file", which for <sourcecode> will be explicitly rendered, as established as best current practice for YANG modules (see for instance RFC 6087 [RFC6087])

Implementation: The current version of xml2rfc uses "name".

This issue is tracked as github issue #36 [2]

3.1.2. In Section 2.12, <br>

A number of elements permits a mixed content model (see Section "Mixed Content Model"): <li>, <blockquote>, <dd>, <td>, and <th>. However, when using the simpler of the two content schemas, two of them (<td> and <th>) permit inline line breaks through the use of <br> elements; the others do not. This seems terribly arbitrary.

Recommendation: Remove the <br> element completely. Alternatively, permit it to be used all places that 'text' and non-block elements may be used (that is, in inline context).

Implementation: The current version of xml2rfc renders <br> as a newline in all inline contexts.

This issue is tracked as github issue #37 [3]

3.1.3. In Section 2.20, <dl>

The current specification says:

> "The "hanging" attribute defines whether or not the term appears on the same line as the definition. hanging="true" indicates that the term is to the left of the definition, while hanging="false" indicates that the term will be on a separate line."

This does not match established typographic terminology. In typographic terminology, "hanging indent" describes the case where the indentation of the second and subsequent lines of a paragraph is greater than the indentation of the first line. Whether the definition in a definition list starts on the first line or not has nothing to do with the presence of hanging indent; our definition lists will _always_ have hanging indent.

The 'hanging' attribute also describes something different from what the term has been used to describe in the version 2 vocabulary. This will be confusing to users.

A more descriptive name for the attribute we're talking about would be 'start-definition-on-first-line', but that's unwieldy. Maybe 'newline="false"' to start the definition on the first line, or something like 'definition-start="first"'?

Recommendation: Change this to a different term that is more descriptive and does not use typographically incorrect terminology.

Implementation: The current version of xml2rfc still uses "hanging".

This issue is tracked as github issue #38 [4]

3.1.4. New Section 2.20.4, "indent" Attribute

The deprecation of the "hangIndent" attribute on <list> leaves no opportunity to control the size of the hanging indent. In some definition lists, it is desirable to have a wide indentation, in

order to clearly show the terms, in other cases it is more important to allow for a larger text volume than the width of the terms would allow.

Recommendation: Add an "indent" attribute on <dl> to control the size of the hanging indent.

Implementation: The current version of xml2rfc does not support the attribute, but has all the underlying functions needed to apply such an attribute. Internally, an indentation is calculated based on length of the <dt> text and the settings of some of the other attributes.

This issue is tracked as github issue #39 [5]

3.1.5. New Section 2.54.2

The version 3 schema deprecates the previously available 'align' attribute for the tables, and the V2 to V3 converter will remove this attributes if used. This makes a previous feature that was appreciated by some authors unavailable. In the text formatter, the effect is simply to make all tables left-aligned, which may not be the most readable and polished output, but for the HTML formatter it also potentially removes the option of letting text flow around smaller tables in a controlled way.

Recommendation: Make the 'align' attribute for tables available again.

Implementation: Implemented but inactive in the current version of xml2rfc. The current text formatter code already has support for the 'align' attribute for these elements; but since the schema does not permit the attribute for <table>, the code is never invoked.

This issue is tracked as github issue #40 [6]

3.1.6. In Section 2.29, <li>

3.1.6.1. Unordered lists with arbitrary symbols

When <li> is used with <ul empty="true">, the rendering is under-specified (the specification says 'no label will be shown", but doesn't say whether list indentation (leading white-space) should be eliminated or not.

If the intention is to make it possible to render unordered lists with arbitrary symbols, chosen on a per-list-item basis, the current

attributes of <li> are insufficient to indent and line-wrap list items properly with <ul empty='true'>.

It is not possible, for instance, to use <ul> lists to generate XML for a table of content, since if the width of the bullet (the section number, in this case) is unknown, the proper indentation and line wrapping cannot be determined.

Recommendation: Add an explicit "bullet" attribute to support this use case.

Implementation: None in the current version of xml2rfc, but internally bullets are taken a configurable bullet list, so accommodating such an attribute would be trivial.

3.1.6.2. Mixed Content Model

The mixed content model for <li> –- either text and inline elements like sub, sup, bcp14, _or_ <t>, <ul>, <figure> etc, is non-intuitive and may be hard for users to keep straight.

Recommendation: Consider simplifying the schema by requiring that text and inline elements always are placed within a <t> element.

Implementation: Not done in the current version of xml2rfc.

This would apply also to other elements that today have alternative content models: <blockquote>, <dd>, <td>, and <th>.

3.1.7. In Section 2.32, <name>

So the <name> element can contain text or <tt>, and <tt> can contain other markup like <sub> and <sup> etc., but why cannot <name> contain <sup> etc. directly?

Recommendation: Change the <name> element schema to permit all inline elements that <tt> can contain, in addition to <tt>.

Implementation: Not changed in the current version of xml2rfc. Implementing this would be a simple matter of changing the v3 schema; no formatter changes would be needed.

3.1.8. In Section 2.40.2, "quoteTitle"

The version two xml2rfc processors already support the attribute "quote-title". The attribute name change introduces an incompatibility. This in particular impacts existing bibxml

reference files, which should work with both version 2 and 3 vocabulary documents.

Recommendation: Change the attribute name back to the value supported by the vocabulary version 2 modes of xml2rfc.

Implementation: The current version of xml2rfc converts "quote-title" to "quoteTitle" during v2v3 conversion, but this is really sub-optimal.

3.1.9. In Section 2.42, <references>

The v3 schema cannot properly model multiple reference subsections contained within one numbered section. The v2 formatter handled this by silently inserting an enclosing section, but with the introduction of the preptool, which in theory should produce a master file from which various formatters would produce equivalent results, this becomes troublesome, as the automatic insertion of a container section is specified for the html formatter, in section 9.8. of RFC 7992, but not for the text formatter. It would be much better to make the prepped xml explicitly show exactly what should be rendered, and not rely on formatters silently insert elements.

Recommendation: Update the schema to make it possible for <references> to contain <references>, and have the prepped xml explicitly show both the encapsulating section and the subsections.

Implementation: Implemented as proposed in the current version of xml2rfc.

3.1.10. In Section 2.45.1, "category" Attribute

Changing the "category" attribute of <rfc> to a name value in an additional <seriesInfo> makes it much harder than it needs to be to look it up. It also makes the semantics of <seriesInfo> less clear.

Recommendation: Remove this, and keep the "category" attribute on <rfc>

Implementation: The "category" attribute on <rfc> has been kept in the current version of xml2rfc, but the additional <seriesInfo> is also generated during v2v3 conversion. For purposes of determining the category to render, the attribute on <rfc> is the one used.

3.1.11. In Section 2.45.3, "docName" Attribute

Changing the "docName" attribute of <rfc> to a name value in an additional <seriesInfo> makes it much harder than it needs to be to look it up. It also makes the semantics of <seriesInfo> even less clear. See also Section 4.4.23.

Recommendation: Remove this, and keep the "docName" attribute on <rfc>

Implementation: The "docName" attribute on <rfc> has been kept in the current version of xml2rfc.

3.1.12. In Section 2.45.7, "number" Attribute

The RFC number attribute in the <rfc> element is used as a switch to control whether an RFC or an Internet-Draft is produced. Moving what is effectively an important controlling switch for the operation of the formatters from the main element down into what is arguably an obscure combination of attribute values on a <seriesInfo> element several levels down from the main element feels wrong.

Recommendation: Don't deprecate the number attribute on <rfc>, but require that the preptool checks that the number attribute matches what's in the <seriesInfo> set. Explicitly mention that the presence of the number attribute on <rfc> causes the generation of an RFC rather than an Internet-Draft by the formatters.

Implementation: In The current version of xml2rfc, the number attribute on <rfc> is used to determine whether to produce an RFC or Internet-Draft. If <seriesInfo> elements are found, but no <seriesInfo> with name="RFC" and value set to the number is found, a warning is given. If no <seriesInfo> elements are found, the appropriate elements, including one giving the RFC number, is inserted.

3.1.13. In Section 2.53.3 and 2.53.4.

3.1.13.1. Unnecessary limitation on where the "keepWithNext" attribute can be used

Why keepWithNext only on <t>? It would be very natural to expect to be able to say keepWithNext for 2 tables, or 2 figures, or 2 lists?

Recommendation: Permit keepWithNext on all elements that can be siblings to <t>.

Implementation: Not in the current version of xml2rfc.

3.1.13.2. Violation of KISS and DRY principles

keepWithNext on one element is equivalent with keepWithPrevious on the following element, provided the following element can have a keepWithPrevious attribute. Providing both violates both KISS [KISS] and DRY (Don't Repeat Yourself) [DRY].

Recommendation: Keep only one of these two attributes, preferably keepWithNext.

Implementation: Not in the current version of xml2rfc.

3.1.14. New Section 2.X, <u>

Thinking about being able to issue warnings both during xml2rfc processing and when running idnits, it seems very hard to distinguish between intentional and non-intentional inclusion of non-ASCII characters in document text.

In addition to the problem of correctly detecting non-intentional use of Unicode characters, there is also the issue (for authors) of correctly converting given Unicode characters to one of the forms recommended in [RFC7997], and the issue (for idnits) of verifying that any Unicode characters or strings are correctly represented as Unicode code-point values next to the literal character or string.

One solution to this could be to not try to guess, or establish heuristics, but instead use a v3 schema element with preptool validation to ensure a straightforward solution to all the issues, as follows:

Proposal: Limit the arbitrary placement of Unicode characters and strings in the body of a document, and control the expansion of the Unicode code-points by requiring that Unicode characters and strings be placed within a specific element if they are to occur in the body of a document. The following text is proposed for inclusion in RFC 7991-bis as a new section:

The <u> element contains a Unicode string which will be rendered according to one of the 6 methods of Unicode renderings listed in [RFC7997], Section 3.4.

In xml2rfc vocabulary version 3, the elements <author>, <organisation>, <street>, <city>, <region>, <code>, <country>, <postalLine>, <email>, and <seriesInfo> may contain non-ascii characters for the purpose of rendering author names, addresses, and reference titles correctly. They also have an additional "ascii" attribute for the purpose of proper rendering in ascii-only media.

However, in order to insert Unicode characters in any other context, xml2rfc vocabulary v3 requires that the Unicode string be enclosed within an <u> element. The element will be expanded inline based on the value of an attribute named "expansion" as follows. Given an element <u expansion="...">Δ</u> in an example sentence:

expansion="numeric-literal":
Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character ("Δ").

expansion="numeric-name":
Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character (INCREMENT).

expansion="numeric-literal-name":
Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character ("Δ", INCREMENT).

expansion="numeric-name-literal":
Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character (INCREMENT, "Δ").

expansion="name-literal-numeric":
Temperature changes in the Temperature Control Protocol are indicated by the "Delta" character "Δ" (U+2206).

expansion="literal-name-numeric":
Temperature changes in the Temperature Control Protocol are indicated by the character "Δ" (INCREMENT, U+2206).

If the <u> element encloses a Unicode string, the rendering reflects this. The element <u expansion="numeric-literal">   </u> will be expanded to 'the characters U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 ("   ")'

Unicode characters which are not enclosed in one of the elements mentioned above will be replaced with a question mark (?) and a warning will be issued.

3.1.15. In Section 2.63.2, <ul> "empty" attribute

In v2, this results in a list using space as the bullet, thus each list entry is indented as with other bullet symbols. However, this leaves no way to get list entries with arbitrary text that are not indented, in order to produce lists such as that used in Table of Content and Index.

Furthermore, the specification does not indicate if <ul empty="true"> should be rendered with space as a bullet, or without any bullet and indentation. A clarification would be good.

Recommendation: Specify that in text output, <ul empty="true"> should be rendered without any bullet and indentation. In order to produce unordered lists that are indented, the "bullet" attribute mentioned in Section 3.1.6 with a white-space bullet could be used.

Implementation: The current version of xml2rfc introduces a new attribute "bare" with the possible values "false" | "true" to signal this. The default is "true" (which differs from the default v2 implementation). Using the extra attribute "bare" works, but is maybe clumsier than necessary.

3.1.16. In Section 3.4.2, "hangIndent" Attribute

> "Deprecated. Use <dl> instead."

This causes capability loss. The "hangIndent" attribute did not only signal that hanging indent should be used, but also gave the size of the indent. No equivalent control has been provided for the <dl> element in the version 3 vocabulary.

Recommendation: Provide an attribute "indent" on <dl> as suggested in Section 3.1.4.

Implementation: Not in the current version of xml2rfc.

3.1.17. In Appendix C. Relax NG schema

The "colspan" attribute is given a default value of "0", this should be "1". "0" is not otherwise defined in the text, and the only reasonable interpretation would be to hide the cell (make it occupy zero columns).

The "rowspan" attribute is given a default value of "0", this should be "1". "0" is not otherwise defined in the text, and the only reasonable interpretation would be to hide the cell (make it occupy zero rows).

Recommendation: Change the default values of "colspan" and "rowspan" to 1.

Implementation: Done in the current version of xml2rfc.

3.1.18. Use of the term "counter".

The classical meaning of this term is a a monotonically increasing sequence of integers, globally unique or unique within a context. In this document, it is instead meant to indicate section, table, figure numbers, which for sections are not plain counters.

To make more interesting, in other contexts in the document, the notation "-nnn", which also would normally indicate a dash followed by digits, i.e., a counter, is also re-interpreted to include section numbers; strings of numbers including embedded period signs. This is bad terminology.

Recommendation: Instead of "counter", use "number" as the attribute value, and explicitly say "Section number, Figure number, Table number or ordered list labels" in the description. Use "-n.n" instead of "-nnn".

Implementation: Not in the current version of xml2rfc.

3.2. RFC 7998

3.2.1. In Section 5.2.6, Attribute Default Value Insertion

The <seriesInfo> "stream" attribute has a default value of "IETF". The effect of setting default values after the XInclude processing is to set stream="IETF" on all reference <seriesInfo> which don't have a stream set. This is probably not right.

Recommendation: Remove the default value for the "stream" attribute from the <seriesInfo> element in the v3 schema.

Implementation: The current version of xml2rfc removes the default value for the "stream" attribute from the schema.

3.2.2. In Section 5.4.6, "pn" Numbering.

The list of elements that are given p- or paragraph tags is severely limited, and since the presence of a pn= attribute is required in order to make internal <xref> instances work, this limits the elements to which it is possible to reference with html fragment identifiers. Why?
Why is <dt> and <li> present, but not <ol>, <dl>, <ul>?

Recommendation: Permit and provide "pn" numbers of type 'paragraph-nnn' for all block-level elements that don't have "pn" numbers otherwise specified.

Implementation: Not in the current version of xml2rfc, but the current version adds p- numbering to <list>, <dl>, <dd>, <ol>, <ul>, which all are allowed to have pn= attributes according to the schema.

4. Non-Schema Issues

4.1. RFC 7991

4.1.1. In Section 2.5.7, "type" Attribute

4.1.1.1. How should a "src" attribute be handled when no "type" is given.

The v3 schema does not require the 'type' attribute on <artwork> to have a value, which makes sense when there's no <artwork> 'src' attribute to include. But if there is a 'src' attribute, but no value for 'type', how should the 'src' value be handled?

The easiest and most explicit handling would be to require a 'type' value if there is a 'src' attribute; a more doubtful alternative would be to use something like the Linux file magic command to try to guess at the content type that 'src' points at.

Recommendation: Warn if there is a 'src' and no 'type' value, and ignore the 'src' in that case.

Implementation: The current version of xml2rfc implements this as proposed.

4.1.1.2. Missing information on how to handle various types

> "The RFC Series Editor will maintain a complete list of the preferred values on the RFC Editor web site, and that list is expected to be updated over time. Thus, a consumer of v3 XML should not cause a failure when it encounters an unexpected type or no type is specified. The table will also indicate which type of art can appear in plain-text output (for example, type="svg" cannot)."

The RFC Series Editor has not yet provided such a table. It is definitely desired, in order to be able to deal correctly with plain-text output.

4.1.2. New Section 2.8.1: Index

There is no guidance on the structure of an index, if one is to be generated by the preptool.

Recommendation: Please provide specification.

Implementation: The current version of xml2rfc provides the generation of index elements in the prepped XML, but makes no claim on the generated XML being optimal.

4.1.3. In Section 2.17, <date>

4.1.3.1. Current Date Requirement

> "When the prep tool is used to create Internet-Drafts, it will reject a submitted Internet-Draft that has a <date> element in the boilerplate for itself that is anything other than today."

It is not up to the format definition to set policy for acceptance or rejection of draft submissions. The matter is more complex than the text assumes, see for instance datatracker issue #2422. In addition to being inappropriate, this text also quietly changes policy from +/- 3 days to +/- 0 days, without saying that it updates RFC 4228 [RFC4228], which is the current specification of permissible dates in draft submissions. Finally, enforcing this would cause _a lot_ of grief and problems.

Recommendation: Remove the section.

Implementation: The current version of xml2rfc does not reject input based on the value of <date>, but warns if the date is more than 3 days from the current date, in accordance with [RFC4228].

4.1.3.2. Date Specification in References

> "Bibliographic references: In dates in <reference> elements, the date information can have prose text for the month or year. For example, vague dates (year="ca. 2000"), date ranges (year="2012-2013"), non-specific months (month="Second quarter"), and so on are allowed."

The text regarding prose text for month and year in bibliographic references is not workable. How should month and year be combined? Some bibliographic references may have date text which requires year first, others year last, and so on. Mixing the described fuzziness into the otherwise strict year, month, date format makes little sense

when the result of combining the year, month and date attributes cannot be predictably and correctly rendered.

Recommendation: Instead of the current specification, permit either that the <date> element may have text content, or an alternative attribute to be used for rendering if year, month, or day cannot be specified exactly.

Implementation: The current version of xml2rfc has not implemented this part of the specification, and is waiting for a more workable solution.

4.1.4. In Section 2.40.1, "anchor" Attribute

Section 5.1 of RFC 7992 says in part:

> "The prep tool produces XML with anchor attributes in all elements that need them."

This is rather vital information regarding the content of the prepped xml when building a formatter, unfortunately it is not mentioned in RFC 7991.

Recommendation: Add this information to the successor of RFC 7991, and to the formatter specifications.

4.1.5. In Section 2.47, <seriesInfo>

The possible and forbidden combinations of attributes for this element has now become so convoluted that it's really hard to understand how to use it correctly. This needs a serious reconsideration.

The 'name' attribute is mandatory, and only 3 values are permitted: "RFC", "Interned-Draft", and "DOI". But it is also mandatory to set the name to "" for a <seriesInfo> with a status attribute. Hmm...

So there are 4, not 3 permitted values: "RFC", "Internet-Draft", "DOI", and "".

This means that all reference files which has things like name="ISO", name="W3C Recommendation", etc., etc., have become illegal.

Recommendation: Do a rewrite of this that does not add new details to the already complex <seriesInfo> semantics, and does not make non-IETF reference files obsolete, but actually simplifies the model and use.

Limit the <seriesInfo> element to what is actually needed for use within <reference/>, and do not add new functionality related to the document <front>. Deprecate any functionality not related to usage within <reference/>.

Implementation: Mostly not done in the current implementation, but see also Section 3.1.10, Section 3.1.12 and Section 3.2.1

4.1.6. In Appendix A.1.1: TLP switch-over date discrepancies

There are discrepancies between the specified switch-over dates in the specification, and those given by the Trust statements:

* TLP3.0: The specification says 2009-11-01 but the TLP statement says effective date 2009-09-12.

* TLP4.0: The specification says 2010-04-01 but the TLP statement says effective date 2009-12-28. The dates on which TLP 4 started to be use in published RFCs seems to match the stated effective date of 2009-12-28, based on a scan of some RFCs around that date.

RFC 7991 also states this about the pre5378 text: this text appears under "Copyright Notice", unless the document was published before November 2009, in which case it appears under "Status of This Memo". This does not agree at all with what actual RFCs contain; they seem to consistently have this text under Copyright Notice.

Recommendation: Correct the dates given in the document to indicate the official dates, and correct the text on placement of TLP to match actual usage.

Implementation: The current version of xml2rfc uses the official dates during the preptool processing, not the dates given in RFC 7991.

4.2. RFC 7992

4.2.1. In Section 8.1.1, Index Contents

The index has an extra <div> enclosing the contents, starting directly after <h2>, while sections explicitly does not have a div here. This irregularity seems quite unnecessary, but makes the formatter code more complex than need be. Could we please align the two?

4.3. RFC 7994

4.3.1. Additional Guidance

* <aside>: Guidance requested on the rendering. Now rendered with an indentation of 9 relative to surrounding text

* <blockquote>: Guidance requested on the rendering. Now rendered with an indentation of 3 spaces, pipe(|), two spaces relative to surrounding text.

* <sub>: Guidance requested. Now rendered as _(text)

* <sup>: Guidance requested. Now rendered as ^(text)

* <tt>: Guidance requested. Now rendered as "text"

* Guidance for <eref> rendering. In the html formatter, handling of <eref> is straightforward and is specified; it simply translates to an external link. In the legacy text formatter, <eref> was handled by inserting an extra <references> subsection called "URLs", and adding reference entries for the URLs there, while the <eref> citation point got a trailing numeric reference number. With the preptool output becoming the authoritative published document, this difference won't be reflected in the xml. The two formats would be more aligned if the text formatter renders <eref> URLs inline.

  Recommendation: Change the rendering of <eref> in text to render the URL inline within parentheses instead of adding the 'URLs' reference subsection.

  Implementation: Implemented in the current version of xml2rfc.

4.4. RFC 7998

4.4.1. In Section 5.2.3, <date> Insertion

Error if any of year, month, day is missing:

It is an unnecessary and unwanted restriction when not in RFC processing mode to given an error for missing date elements. Missing date elements have been permitted because they make it easier for draft authors to rev drafts without having to pay attention to the date values every time they generate new output. This requirement should apply only to RFC prepping mode, and only in part:

In RFC processing mode, this implicitly changes the RFC-Editor policy regarding publication dates, which earlier have specified only year and month (except for April 1st RFCs). Is this intentional?

Recommendation: Remove this restriction for draft mode, and modify it to require only year and month in RFC mode.

Implementation: The current version of xml2rfc warns if not all three elements are present in RFC mode. The tool author considers even this inappropriate.

In Internet-Draft mode, the current implementation handles missing elements the same way that the v2 formatters do.

4.4.2. In Section 5.2.4, "prepTime" Insertion

This is under-specified, given the detailed requirements on the <date> attributes. Should probably be specified as format according to [RFC3339], with year, month, day, hour, minute, and second.

Recommendation: Specify the format as RFC3339 compliant with resolution at least down to a second.

Implementation: Implemented as RFC3339 with year, month, and day up to version 2.10.3; changed to the proposal above in the next release.

4.4.3. In Section 5.2.6, Attribute Default Value Insertion

All the default values in 7991 are also expressed in the v3.rnc schema. Remove text indicating otherwise. And by the way, it was very helpful to extract these from the schema programmatically; having them specified otherwise would make it much harder to follow a changing schema.

A number of attributes which are deprecated have default values. The current specification will cause those to be inserted, even if they have been removed earlier by the v2v3 converter because they are deprecated. This seems inconsistent.

Recommendation: Omit deprecated attributes from the default-setting.

Implementation: Not in the current version of xml2rfc.

4.4.4. In Section 5.2.7, "toc" Attribute

It's specified that sections with <boilerplate> ancestors should have toc="exclude", but this won't then affect <boilerplate> sections which are inserted as part of the processing in 5.4.2. It would make more sense to move this processing to after 5.4.2.

The logic in the second bullet is flawed. First it says to set elements with children with toc="include" to "include", but then it says that it is an error if they are set to "exclude". Either there should be a warning, and the toc= attribute should be updated, or there should be an error and termination. Not both.

Recommendation: Move 5.2.7 processing to after 5.4.2, or specify that a second pass should be done after boilerplate insertion. If a parent to a section with toc="include" has toc="exclude", an error should be generated.

Implementation: In order to do the actions of 5.2.7 for boilerplate, a second pass is made after boilerplate insertion in the current version of xml2rfc. Handling of inconsistent "toc" attribute settings is implemented as proposed.

4.4.5. In Section 5.2.8, "removeInRFC" Warning Paragraph

This potentially inserts a new <t> element, but after the default setting in 5.2.6.

Recommendation: Maybe place default setting after all potential element insertions have taken place.

Implementation: The current version of xml2rfc deals with this by adding default-setting of attributes individually on each new elements as they are inserted. This works, but is more complex and probably less efficient than doing default-setting once, after any new elements have been inserted.

4.4.6. In Section 5.3.1, "month" Attribute

> "Normalise the values of "month" attributes in all <date> elements in <front> elements in <rfc> elements to numeric values."

Is that 'in' a direct descendant relationship, or any descendant? I.e., does this affect <date> elements in included <reference> elements? Unclear. (RFC7991 is much clearer on this point, but that's not an excuse for being unclear here).

Recommendation: Clarify the text.

4.4.7. In Section 5.3.2, ASCII Attribute Processing

The uppercasing of 'ascii' in the section <name> is incorrect in this case; the attribute name is explicitly 'ascii', not 'ASCII'. The section name should be '"ascii" Attribute Processing'.

Recommendation: Change the title 'ASCII Attribute Processing' to refer correctly to the "ascii" attribute: '"ascii" Attribute Processing'.

> "In every <author> element ..."

After the earlier XInclude processing, this will include all the author elements in the included references, which the document author should not normally change in any way. Was this the intention?

Recommendation: Limit it to /rfc/front/author' elements.

Implementation: Implemented in the current version of xml2rfc.

<title> and <postalLine> also has an "ascii" attribute — is it a mistake that they are not mentioned here? Assuming so, for the preptool implementation.

What about the ascii* attributes on author? Assuming they should be processed the same way.

Recommendation: Process all "ascii" attributes in the document <front> as specified, and ignore those within <references>

Implementation: Implemented as proposed.

4.4.8. New Section 5.3.4: "keepWithNext" Normalisation

Recommendation: The new section should specify normalisation of keepWithNext/keepWithPrevious such as to replace all keepWithPrevious with an equivalent keepWithNext on the previous element, in case the proposal in Section 3.1.13.2 is not accepted.

Implementation: Not in the current version of xml2rfc.

4.4.9. In Section 5.4.2, <boilerplate> Insertion: Only for RFCs?

> "Create a <boilerplate> element if it does not exist. If there are any children of the <boilerplate> element, produce a warning that says "Existing boilerplate being removed. Other tools, specifically the draft submission tool, will treat this condition as an error" and remove the existing children."

Should this be done in both I-D mode and RFC mode? The trouble is that the following subsections only describes the boilerplate relevant to an RFC; there's additional boilerplate that is needed for drafts. I don't think it's reasonable to have a draft with only parts of the boilerplate contained in a boilerplate section.

Recommendation: The boilerplate-element insertion parts of 5.4.2 should be done in both RFC and draft mode, with the appropriate boilerplate for each case. For consistency, either add text to describe the appropriate boilerplate for drafts, or remove the sections specific to RFC boilerplate.

Implementation: The current version of xml2rfc inserts boilerplate for both drafts and RFCs, as appropriate.

4.4.10. In Section 5.4.2, <boilerplate> Insertion: Error Message

This section also specifies an error message to be used verbatim; the troublesome thing is that it's not clear what it means. The message is: "Existing boilerplate being removed. Other tools, specifically the draft submission tool, will treat this condition as an error". What is it that the draft submission tool is going to treat as an error? The presence of boilerplate? Why? The removal of boilerplate? How is that related to draft submission? This is very jumbled.

Recommendation: If existing boilerplate is found, issue a warning and replace it.

For other tools, suggest that if boilerplate is present during draft submission, it should be checked for validity. This is already a function of idnits, so does not constitute anything new, but is decidedly better than having the submission tool actually reach into the submitted document and change it.

Implementation: In the current version of xml2rfc this is implemented as proposed, with the following warning if existing boilerplate is found: "Expected no <boilerplate> element, but found one. Replacing the content with new boilerplate."

4.4.11. In Section 5.4.2.1, Compare <rfc> submissionType and <seriesInfo> "stream".

This comes too late. It is specified that if either is missing, it should be added. But the default attribute setting earlier has set stream="IETF" on all <seriesInfo> elements that didn't have it. If a document is read without submissionType, and stream set correctly to something else than "IETF" on one of the <seriesInfo> elements, then

the default-setting will have created a conflict which cannot be resolved purely from the document at this point.

Furthermore, it doesn't seem like a good fit to have tag attributes that all have to be set to the same value. This is not DRY, and unnecessarily introduces the possibility of conflict, as a result of multiple <seriesInfo> elements being permitted (Relevant to the v3 schema, not the preptool).

Recommendation: Remove the default value for stream, and make it subordinate to submissionType.

Implementation: The current version of xml2rfc implements the specification as written, and produces errors (which lead to not producing an output document) on inconsistencies. This does not feel user-friendly.

4.4.12. In Section 5.4.2.2, "Status of this Memo" Insertion

It specifies that one should consider both submissionType and <seriesInfo> stream value; but those have just been set equal in 5.4.2.1.

Recommendation: Remove <seriesInfo> from consideration here. In order to produce a correct "Status of this Memo" text, "category", "consensus", and "submissionType" must be considered, and all three are present as attributes on <rfc>. Keep it that way.

Implementation: The current version of xml2rfc looks at "submissionType", "category", and "consensus" on the <rfc> element.

4.4.13. In Section 5.4.3, <reference> "target" Insertion

> "Insert "target" attributes for RFC, DOI, and Internet-Draft references that lack them."

It is indicated that the rfc-editor will provide the URL patterns. What are they?

In the formatter, the order of <seriesInfo> determines the rendering order. The insertion should probably be done in the desired rendering order.

Recommendation: In addition to providing the appropriate URL patterns, specify the order in which the <seriesInfo> elements should occur, for instance: 'BCP', 'RFC', 'DOI'.

Implementation: The current version of xml2rfc inserts the appropriate <seriesInfo> elements, and after insertion sorts them in the order 'BCP', 'RFC', 'DOI', followed by others.

4.4.14. In Section 5.4.4, <name> Slugification

The 'n-' prefix for slugs is unnecessarily opaque.

Recommendation: Use slugs with prefix "name-" rather than "n-", to be more self-documenting.

Implementation: Implemented as proposed in the current version of xml2rfc.

Should the slugs be unique? Assuming yes, but guidance would be good. The current version of xml2rfc enforces unique slugs, with the following algorithm:

* remove non-ascii letters

* replace-non-letters with dash, compacting multiple dashes to one

* reduce length to 32, but insure uniqueness by increasing length or adding numerical suffixes, up to length 40 with suffixes numbered 2 to 99.

Recommendation: Do slugification and uniqueness enforcement as described above.

Implementation: As described above.

4.4.15. In Section 5.4.6, "pn" Numbering.

What does 'pn' mean? Cryptic is never good when humans have to deal with it. At least explain as "part number" in text. Possibly even change pn="" to part="".

<back><section> is not mentioned. Assuming numbering as section-appendix.1.2

<iref> elements are not mentioned (but covered in 7991). Should be listed in 7998.

The numbering scheme is inconsistent between notes/boilerplate and other sections, in that if attempting to split a pn on dashes (which external tools might want to do) the boilerplate/note sections contain an additional dash.

Recommendation: Change that dash to a dot, for better consistency with other sections. This also makes the <t> part numbers less confusing: "section-boilerplate.1-1" instead of "section-boilerplate-1-1"

Implementation: Implemented as proposed in the current version of xml2rfc.

4.4.15.1. RFC format anchors / fragment identifiers

The anchor prefixes described unnecessarily break with existing links to document sections. Wikipedia has (2018-02-19) about 84 000 pages that link to RFCs; with most pages having multiple links. A small manual sampling indicates that about 1 link in 10 has a #section-fragment identifier. All of these will break if the new tools are used to generated content linked from these pages.

How much larger than Wikipedia is the whole of the internet, in terms of links to RFCs? Hard to tell (though searching for 'rfc' on Google indicates 'about 10 000 000 results). In any case, we are talking about breaking a substantial number of links using fragment identifiers of the format #section- and #appendix- if the new tools are used to replace the old html content that sites currently point to.

Recommendation: Update the RFC 7998 preptool to use these prefixes, instead:

- "section-xxx"
- "figure-xxx"
- "table-xxx"
- "appendix-xxx"
- "index-xxx"
- "para-xxx"
- "name-xxx"

Implementation: Implemented as above in the current version of xml2rfc.

4.4.16. In Section 5.4.7, <iref> Numbering

Numbering of <iref> talks about setting the 'pn' attribute. Mixed into this is a mention of 'irefid', which isn't a valid attribute. The current implementation assumes that 'pn' is meant.

The item and sub-item text is not constrained to slug format; in order to deliver useful pn values, slugification should be done. On the other hand, the explicit prescription of how to ensure uniqueness clashes with the total lack of uniqueness attention under 5.4.4.

Recommendation: Require slugification for pn-numbering of items and sub-items, but remove the details of how to ensure uniqueness. Correct the mention of 'irefid' to say 'pn', if that was intended.

Implementation: Slugification is done, and uniqueness is enforced with an algorithm that limits slug length and tries to keep slugs readable. If there are more than 99 slugs that would collide if no uniqueness processing was done, an error is generated.

4.4.17. In Section 5.4.8.2, "derivedContent" Insertion (without Content)

There's a formatting mistake:

The last sentence of the last bullet ("Issue a warning...") should not be part of the bullet, but a separate final paragraph for the Section.

4.4.18. In Section 5.5.1, <artwork> Processing

RFC791 specifies that the <artwork> content is a fallback if there is external <svg> content, but 7998 says to drop the fallback and insert the external <svg>. This deletes information, and makes the fallback unavailable. This needs a better handling.

Recommendation: If there is fallback content, convert the external URL content to a "data:" URL for the src. This pulls the external content in and makes it immutable, but retains the fallback text.

Implementation: Implemented as proposed in the current version of xml2rfc.

4.4.19. In Section 5.5.2, <sourcecode> Processing

List item 4 says:

"fill the content of the <sourcecode> element with the resolved XML from the URI in the "src" attribute"

However, we have no particular reason to assume that the content of the "src" URL is XML. Quite to the contrary, it would be a very natural and common use case that the external content is a source code file.

Recommendation: The URI should not be assumed to resolve to xml, but instead treated like CDATA.

Implementation: Implemented as proposed in the current version of xml2rfc.

4.4.20. In Section 5.4.8.2, "derivedContent" Insertion.

It is not clear from the description if the derived content text should contain square brackets when an <xref> would be rendered with square brackets in current output formats.

It is not clear if the derived content should include the 'Figure', or 'Table' label when pointing to such objects. When rendering such a reference in the current output formats, the generated text would include the label, but the current text seems to lean towards not making this part of the derived content, which would cause incompatibility with the output of v2 formatters.

The purpose of this is insufficiently explained. If the intention is to use this when generating derived formats, there are problems: If, for instance, the derived format with a <reference> target is set to 'RFC1234', the text inserted in a derived format should have surrounding square brackets; but if the target is a section, it should not. If on the other hand the derived format includes the square brackets when appropriate, the link in a derived format with internal link capability will use the whole of the bracketed string, rather than the more appropriate text within the brackets.

Recommendation: The whole "derivedContent" handling and specification needs a thorough rework, with specification of the intended use of the attribute by formatters. Possibly the whole "derivedContent" concept should be scrapped, and the rendering left for the formatter, depending on the characteristics of the output format.

Implementation: The current version of xml2rfc works around this issue by using different formatter code for different cases, which is not good from the viewpoint of using the prepped XML as the archival format, but at least produces reasonable output.

4.4.21. In Section 5.4.9, <relref> Processing

Why doesn't <relref> have the same format options as <xref>? Surely they must be just as relevant here. But more importantly, <relref> overlaps <xref> so much that it would be better to just add section, relative, and displayFormat to <xref>. Maybe change displayFormat to the earlier proposed 'sectionFormat'.

Recommendation: Deprecate <relref>, and fold the functionality into <xref>.

Implementation: The <relref> functionality has been folded into <xref>, but relref support not yet removed.

4.4.22. New Section 5.4.10, Index Insertion

RFC7998 does not say anything about inserting xml for the index, if one is requested, but it seems counter-intuitive not to produce xml for the index as part of the preptool processing, given all the other prepping that's being done. What's more, in Section 2.27 of RFC 7991 there's this text:

> "When the prep tool is creating index content, it collects the items in a case-sensitive fashion for both the item and sub-item level."

Recommendation: Insert the XML necessary to render the index into the prepped XML.

Implementation: Implemented as proposed in the current version of xml2rfc.

4.4.23. In Section 5.6.3, <link> Processing

Bullet 4.: Bad grammar: s/RFC the form/RFC, in the form/

Bullet 4.: Hmm. The <link rel="convertedFrom" href="draft-..."> should ideally be created automatically, but there is no clear path of how to do that.

Recommendation: Require docName to be set to the draft name, and use that to create this link. This also implies that "docName" not be deprecated (see Section 3.1.11).

Implementation: Implemented as proposed in the current version of xml2rfc.

5. Security Considerations

This document does not introduce any security considerations on its own.

6. References

6.1. Informative References

[DRY] Wikipedia, "Don't repeat yourself", 2018, <https://en.wikipedia.org/wiki/Don%27t_repeat_yourself>.

[KISS] Wikipedia, "KISS Principle", 2018, <https://en.wikipedia.org/wiki/KISS_principle>.

[RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <https://www.rfc-editor.org/info/rfc3339>.

[RFC4228] Rousskov, A., "Requirements for an IETF Draft Submission Toolset", RFC 4228, DOI 10.17487/RFC4228, December 2005, <https://www.rfc-editor.org/info/rfc4228>.

[RFC6087] Bierman, A., "Guidelines for Authors and Reviewers of YANG Data Model Documents", RFC 6087, DOI 10.17487/RFC6087, January 2011, <https://www.rfc-editor.org/info/rfc6087>.

[RFC7749] Reschke, J., "The "xml2rfc" Version 2 Vocabulary", RFC 7749, DOI 10.17487/RFC7749, February 2016, <https://www.rfc-editor.org/info/rfc7749>.

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016, <https://www.rfc-editor.org/info/rfc7991>.

[RFC7992] Hildebrand, J., Ed. and P. Hoffman, "HTML Format for RFCs", RFC 7992, DOI 10.17487/RFC7992, December 2016, <https://www.rfc-editor.org/info/rfc7992>.

[RFC7993] Flanagan, H., "Cascading Style Sheets (CSS) Requirements for RFCs", RFC 7993, DOI 10.17487/RFC7993, December 2016, <https://www.rfc-editor.org/info/rfc7993>.

[RFC7994] Flanagan, H., "Requirements for Plain-Text RFCs", RFC 7994, DOI 10.17487/RFC7994, December 2016, <https://www.rfc-editor.org/info/rfc7994>.

[RFC7995] Hansen, T., Ed., Masinter, L., and M. Hardy, "PDF Format for RFCs", RFC 7995, DOI 10.17487/RFC7995, December 2016, <https://www.rfc-editor.org/info/rfc7995>.

[RFC7996] Brownlee, N., "SVG Drawings for RFCs: SVG 1.2 RFC", RFC 7996, DOI 10.17487/RFC7996, December 2016, <https://www.rfc-editor.org/info/rfc7996>.

[RFC7997] Flanagan, H., Ed., "The Use of Non-ASCII Characters in RFCs", RFC 7997, DOI 10.17487/RFC7997, December 2016, <https://www.rfc-editor.org/info/rfc7997>.

[RFC7998] Hoffman, P. and J. Hildebrand, ""xml2rfc" Version 3 Preparation Tool Description", RFC 7998, DOI 10.17487/RFC7998, December 2016, <https://www.rfc-editor.org/info/rfc7998>.

[XML2RFC] Levkowetz, H., "xml2rfc", 2018, <https://pypi.org/pypi/xml2rfc>.

6.2. URIs

[1] https://pypi.org/pypi/xml2rfc

[2] https://github.com/rfc-format/draft-iab-xml2rfc-v3-bis/issues/36

[3] https://github.com/rfc-format/draft-iab-xml2rfc-v3-bis/issues/37

[4] https://github.com/rfc-format/draft-iab-xml2rfc-v3-bis/issues/38

[5] https://github.com/rfc-format/draft-iab-xml2rfc-v3-bis/issues/39

[6] https://github.com/rfc-format/draft-iab-xml2rfc-v3-bis/issues/40

Author's Address

Henrik Levkowetz
Elf Tools AB
Ollonstigen 8
Sweden

Email: henrik@levkowetz.com