

Workgroup: Network Working Group
Internet-Draft:
draft-levkowetz-xml2rfc-v3-implementation-
notes-12
Published: 15 December 2020
Intended Status: Informational
Expires: 18 June 2021
Authors: H. Levkowetz
Elf Tools AB

**Implementation notes for RFC7991,
"The 'xml2rfc' Version 3 Vocabulary"**

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

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 18 June 2021.

Copyright Notice

Copyright (c) 2020 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

- [1. Introduction](#)
 - [1.1. Current Status](#)
- [2. Fitness for Purpose](#)
 - [2.1. Degraded Table of Contents](#)
 - [2.2. RFC Publication Date Policy](#)
- [3. Schema Issues](#)
 - [3.1. RFC 7991](#)
 - [3.1.1. Before Section 2.5: <artset>](#)
 - [3.1.2. In Section 2.5.5, "name" Attribute](#)
 - [3.1.3. In Section 2.5.7, <artwork> "type" Attribute](#)
 - [3.1.4. In Section 2.6, <aside>](#)
 - [3.1.5. In Section 2.12,
](#)
 - [3.1.6. In Section 2.20, <dl>](#)
 - [3.1.7. New Section 2.20.4, "indent" Attribute](#)
 - [3.1.8. New Section 2.54.2](#)
 - [3.1.9. In Section 2.27, <iref>](#)
 - [3.1.10. In Section 2.29,](#)
 - [3.1.11. In Section 2.32, <name>](#)
 - [3.1.12. In Section 2.32, <organization>](#)
 - [3.1.13. In Section 2.37, <postal>](#)
 - [3.1.14. In Section 2.40.2, "quoteTitle"](#)
 - [3.1.15. In Section 2.41, <referencegroup>](#)
 - [3.1.16. In Section 2.42, <references>](#)
 - [3.1.17. In Section 2.45.1, "category" Attribute](#)
 - [3.1.18. In Section 2.45.3, "docName" Attribute](#)
 - [3.1.19. In Section 2.45.7, "number" Attribute](#)
 - [3.1.20. In Section 2.46.2, "numbered" Attribute](#)
 - [3.1.21. In Section 2.47, <seriesInfo>](#)
 - [3.1.22. In Section 2.48, <sourcecode>](#)
 - [3.1.23. In Section 2.53.3 and 2.53.4.](#)
 - [3.1.24. New Section 2.X, <u>](#)
 - [3.1.25. In Section 2.63.2, "empty" attribute](#)
 - [3.1.26. In Section 2.66.1, <xref> "format" attribute](#)
 - [3.1.27. In Section 3.3, <format>](#)
 - [3.1.28. In Section 3.4.2, "hangIndent" Attribute](#)
 - [3.1.29. In Appendix C. Relax NG schema](#)
 - [3.1.30. Use of the term "counter".](#)
 - [3.1.31. In Section 2.44, <relref>](#)
 - [3.1.32. In Section 2.66, <xref>](#)
 - [3.1.33. Contributor names](#)
 - [3.2. RFC 7998](#)
 - [3.2.1. New Section 5.1.6, Attribute validation](#)
 - [3.2.2. In Section 5.2.6, Attribute Default Value Insertion](#)
 - [3.2.3. In Section 5.4.6, "pn" Numbering.](#)
 - [3.3. Some attributes should have value type xsd:ID](#)

4. Non-Schema Issues

4.1. RFC 7991

- 4.1.1. In Section 2.5.7, "type" Attribute
- 4.1.2. New Section 2.8.1: Index
- 4.1.3. In Section 2.17, <date>
- 4.1.4. In Section 2.40.1, "anchor" Attribute
- 4.1.5. In Section 2.48.4, "type" Attribute
- 4.1.6. In Appendix A.1.1: TLP switch-over date discrepancies
- 4.1.7. In Appendix B.2.1: Generation of PN numbers

4.2. RFC 7992

- 4.2.1. In Section 5.1, IDs
- 4.2.2. In Section 6.2, Root Element
- 4.2.3. In Section 6.4, Page Headers and Footers
- 4.2.4. In Section 6.5, Document Information
- 4.2.5. In Section 8.1.1, Index Contents
- 4.2.6. Inconsistent use of "s-", "n-" and User-Supplied "id" Attributes
- 4.2.7. In Section 9.2, <address>
- 4.2.8. In Section 9.7.2, Authors of this Document
- 4.2.9. In Section 9.7.3, Authors of References
- 4.2.10. In Section 9.16, <cref>
- 4.2.11. In Section 9.24, <eref>
- 4.2.12. In Section 9.25, <figure>
- 4.2.13. In Section 9.27, <iref>
- 4.2.14. In Section 9.33, <note>
- 4.2.15. In Section 9.34,
- 4.2.16. In Section 9.35, <organization>
- 4.2.17. In Section 9.36, <phone>
- 4.2.18. In Section 9.37, <postal>
- 4.2.19. In Section 9.40, <reference>
- 4.2.20. In Section 9.41, <referencegroup>
- 4.2.21. In Section 9.42, <references>
- 4.2.22. In Section 9.54, <table>
- 4.2.23. In Section 9.56, <td>
- 4.2.24. In Section 9.58, <th>
- 4.2.25. In Section 9.60, <title>
- 4.2.26. In Section 9.66, <xref>
- 4.2.27. In Section 9.18, <dd>

4.3. RFC 7994

- 4.3.1. Additional Guidance

4.4. RFC 7998

- 4.4.1. In Section 5.2.3, <date> Insertion
- 4.4.2. In Section 5.2.4, "prepTime" Insertion
- 4.4.3. In Section 5.2.6, Attribute Default Value Insertion
- 4.4.4. In Section 5.2.7, "toc" Attribute
- 4.4.5. In Section 5.2.8, "removeInRFC" Warning Paragraph
- 4.4.6. In Section 5.3.1, "month" Attribute
- 4.4.7. In Section 5.3.2, ASCII Attribute Processing
- 4.4.8. New Section 5.3.4: "keepWithNext" Normalisation

- [4.4.9. In Section 5.4.2, <boilerplate> Insertion: Only for RFCs?](#)
- [4.4.10. In Section 5.4.2, <boilerplate> Insertion: Error Message](#)
- [4.4.11. In Section 5.4.2.1, Compare submissionType and <seriesInfo> "stream".](#)
- [4.4.12. In Section 5.4.2.2, "Status of this Memo" Insertion](#)
- [4.4.13. In Section 5.4.3, <reference> "target" Insertion](#)
- [4.4.14. In Section 5.4.4, <name> Slugification](#)
- [4.4.15. In Section 5.4.6, "pn" Numbering.](#)
- [4.4.16. In Section 5.4.7, <iref> Numbering](#)
- [4.4.17. In Section 5.4.8.1, "derivedContent" Insertion \(with Content\)](#)
- [4.4.18. In Section 5.4.8.2, "derivedContent" Insertion \(without Content\)](#)
- [4.4.19. In Section 5.5.1, <artwork> Processing](#)
- [4.4.20. In Section 5.5.2, <sourcecode> Processing](#)
- [4.4.21. In Section 5.4.8.2, "derivedContent" Insertion.](#)
- [4.4.22. In Section 5.4.9, <relref> Processing](#)
- [4.4.23. New Section 5.4.10, Unused Reference Warnings](#)
- [4.4.24. New Section 5.4.11, Index Insertion](#)
- [4.4.25. In Section 5.6.3, <link> Processing](#)
- [5. Possible New Work](#)
 - [5.1. Inline and Display Math](#)
 - [5.2. Change Bars](#)
 - [5.3. Element Nesting](#)
 - [5.4. Schema Consistency](#)
- [6. Security Considerations](#)
- [7. Informative References](#)
- [Appendix A. Proposed new sections in RFC 7991 bis](#)
 - [A.1. <u>](#)
 - [A.1.1. Expansion of simplified <u> format specifications](#)
 - [A.1.2. Non-simplified <u> format specifications](#)
 - [A.1.3. Split expansion of <u> elements](#)
 - [A.2. <rendering>](#)
- [Author's Address](#)

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](#) at the date this document was published, as given above.

1.1. Current Status

For most of the issues listed in this document, a resolution is now (14 Jul 2020, draft-levkowetz-xml2rfc-v3-implementation-notes-11) available. For most issues where the resolution imply changes compared with the published specifications, the changes have been made over time, and are available in the released xml2rfc. However, some issues remain:

*Separation of type and content for <artwork>, [Section 3.1.3](#):
The implementation only recognises type values that are related to the format of the artwork, for instance "svg" and "ascii-art", but no attribute has been designated to hold content types, such as "call-flow". We can either simply narrow the description of the "type" attribute of <artwork>, and eliminate the support for content labelling, or add a new attribute.

*Introduction of a "bullet" attribute for [Section 3.1.10.1](#).
This would let us get rid of the awkward "bare" attribute that was introduced to make unindented lists possible, something that was needed by the RPC; and would be a much more general solution.

*Simplification of mixed-content elements, [Section 3.1.10.2](#).
Confusion about when you have to use a sequence of <t> elements within for instance elements, and when you can use bare text, keeps coming up as a point of confusion from users on the mailing lists. Being able to omit the wrapping of text in <t> in

some cases is a small convenience that doesn't make up for the confusion caused by the mixed content model.

*Deprecation of the new "quoteTitle", keep original "quote-title", [Section 3.1.14](#).

*Official deprecation of the new <seriesInfo> attributes, [Section 3.1.21](#). Implementing this, and reverting to v2 behaviour would permit better error messages and would simplified documentation and usage no end. Currently <seriesInfo> instances in references are moved fr

*Permitting "keepWithNext" on all elements that can be siblings to <t>, in order to make it useful also for other child elements of <section>, [Section 3.1.23.1](#). Not being able to set "keepWithNext" on other child elements of <section> than <t> repeatedly comes up as an issue preventing better page break handling when generating PDFs. (There are also issues with the WeasyPrint engine in this area, but not having general "keepWithNext" support makes the issue harder than need be).

*Permitting an "asciiAbbrev" attribute for <title>, [Section 3.1.12.1](#), to match the "ascii" attribute for the non-abbreviated title.

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.

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

Implementation: The current implementation provides pagination for drafts. The pagination can be turned off with an option switch. Text/plain output for RFCs is always generated without pagination.

Heather's indication 20 Jul 2019: OK for IDs but default should not change for RFCs

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.

Proposal: 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: The current version of xml2rfc does not enforce the requirement that all three <date> elements are present in RFC mode, but leaves up to author and RPC (RFC Production Center) staff to insert day information as appropriate.

Heather's indication 20 Jul 2019: Desired behaviour is to be able to publish with or without exact date.

3. Schema Issues

3.1. RFC 7991

3.1.1. Before Section 2.5: <artset>

The way <artwork> has been specified to handle the presence of both SVG artwork and text fallback (in Section 2.5 of [[RFC7991](#)]) has the result that any SVG content has to be placed as a data: URL in the

"src" attribute when an ascii-art fallback is present. This makes the SVG effectively uneditable once the preptool has been run, even if the SVG artwork was originally provided as a regular SVG XML file external to the document XML file.

In order to be able to more easily deal with alternative instances of artwork, and in the future possibly deal smoothly with a wider number of alternative artwork formats than is currently provided for, a new element <artset> could be introduced, presenting a set of alternative artwork executions. This would let the renderer pick the most appropriate <artwork> instance for its format from the alternatives present within an <artset> element, based on the "type" attribute of each enclosed <artwork> element.

If more than one <artwork> element is found within an <artset> element, with the same "type" attribute, the renderer could select the first one, or possibly choose between the alternative instances based on the output format and some quality of the alternative instances that made one more suitable than the other for that particular format, such as size, aspect ratio, or whatnot.

Implementation: Xml2rfc as of version 2.19.0 implements this, with a preference list when rendering to HTML and PDF of ("svg", "binary-art", "ascii-art"), while the text renderer uses the list ("ascii-art",) -- i.e., one entry only. The Relax-NG compact schema used for <artset> is this:

```
artset =
  element artset {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { xsd:ID }?,
    artwork+
  }
```

The <artset> element can occur anywhere an <artwork> element can occur. The first anchor on an <artwork> element within an <artset> element will be promoted to the <artset> element if it has none; apart from that, anchors on <artwork> elements within an <artset> element will be removed by the preptool.

Heather's indication 20 Jul 2019: OK

3.1.2. 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.

Proposal: 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 as specified in [[RFC8407](#)].

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

Resolution: The attribute "name" was used for this purpose already in v2 of the vocabulary. Closed with no action.

Heather's indication 20 Jul 2019: OK on resolution

This issue is tracked as [github issue #36](#)

3.1.3. In Section 2.5.7, <artwork> "type" Attribute

The text lists a number of preferred values, but does not indicate how these are to be used, or what to do with other values. In particular, the default value is "" (i.e., empty) -- should this cause a warning or error, or any other action? If not, how should 'preferred' be understood?

Additionally, according to Section 5.1 of RFC 7991, any text content serves as ascii-art fallback in case the rendering format cannot render the content that the 'src' attribute indicates. But in that case, it seems that the "type" attribute should apply exclusively to the content that the "src" attribute points at. This should be clarified in the text.

Further, some thought about the possible use cases for the listed preferred values of the "type" attribute makes it appear that the given list contains values from (at least) two different classes of things:

- *"svg" seems to describe a format

- *"binary-art" also seems to describe a format

- *"ascii-art" also seems to describe a format

- *"call-flow" seems to describe the art content

- *"hex-dump" seems to describe the art content

Proposal: Require the "type" attribute to have a value if the "src" attribute is specified, and let it describe the format. If any action should be taken on the basis of one of the preferred

values appearing or a different value appearing, add text to indicate so.

For values like "call-flow" and "hex-dump", add a different attribute to describe the artwork content. Do not conflate the artwork description with the artwork format given in the "type" attribute.

Implementation: The current implementation uses the "type" attribute to determine how to process the "src" attribute. Handling exists for the values "svg", "binary-art", and "ascii-art". The idnits rewrite warns if type has any value other than "svg", "binary-art", or "ascii-art".

As of version 2.19.0 of xml2rfc, the conflict between the type that the "src" attribute points at, and any ascii-art fallback has been removed by introduction of the <artset> element. A solution is still needed if it's desired to have an attribute that describes the content type.

Heather's indication 20 Jul 2019: OK on the implementation (accepting only format types, not content types, for the "type" attribute, and using that to determine which <artwork> to select).

3.1.4. In Section 2.6, <aside>

3.1.4.1. Child element <list>

The schema permits <list> inside <aside>, but <list> is deprecated, and <aside> is a new vocabulary v3 element, so they should never be able to occur together, it seems to me.

Proposal: Don't permit <list> inside <aside>.

Implementation: Implemented in the current version of xml2rfc.

Heather's indication 20 Jul 2019: OK

3.1.4.2. Child element <table>

The schema permits <table> inside <aside>, but does not permit <table> inside <blockquote>. Lacking any indication of why this is, it seems reasonable to propose that the schema be adjusted to permit <table> inside either both or neither.

An added consideration is that appropriate rendering of table headers and footers across page breaks may be in conflict with rendering of <table> within <aside> and <blockquote>.

Implementation:

The current implementation (xml2rfc version 2.21.x) permits `<table>` inside both `<aside>` and `<blockquote>`, but does not guarantee that `<aside>`s and `<blockquote>`s broken across pages will have new table headers and footers added if a table inside is split over multiple pages.

Heather's indication 20 Jul 2019: OK

3.1.5. In Section 2.12, `<br>`

A number of elements permits a mixed content model (see [Section 3.1.10.2](#)): `<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.

Proposal: 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).

Resolution: After repeated list discussion, the `<br>` element was accepted in inline context.

The implementation permits this element as a child element of `blockquote`, `cref`, `dd`, `dt`, `em`, `li`, `name`, `strong`, `t`, `td`, `th`, `title`, and `tt`.

This issue is tracked as [github issue #37](#)

3.1.6. 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"'?

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

Resolution: The "hanging" attribute will be renamed to "newline", with newline="true" meaning the same as hanging="false". The default value will change accordingly.

This issue is tracked as [github issue #38](#)

3.1.7. 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.

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

Resolution: An "indent" attribute will be added on <dl> to control the size of the hanging indent. The value will signify the number of character positions in text/plain rendering, and a count of 0.5em distances in richer renderings.

Heather's indication on 20 Jul 2019: OK

This issue is tracked as [github issue #39](#)

3.1.8. 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.

Proposal:

Make the 'align' attribute for tables available again.

Resolution: An attribute "align" will be re-introduced for table alignment, with the possible values "left", "center", and "right".

Heather's indication 20 Jul 2019: OK

This issue is tracked as [github issue #40](#)

3.1.9. In Section 2.27, <iref>

In HTML5, may not be placed directly inside a table. RFC 7992 specifies that <iref> should be rendered as a , and also specifies that <table> is directly rendered as its HTML counterpart. This results in generating invalid HTML.

Proposal: Disallow <iref> as a direct child of <table> (but still permitting it within <th> and <td>).

Implementation: The current implementation works around this by moving the outside the <table>. This is less than ideal.

**3.1.10. In Section 2.29, **

3.1.10.1. Unordered lists with arbitrary symbols

When 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 whitespace) 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 are insufficient to indent and line-wrap list items properly with <ul empty='true'>.

It is not possible, for instance, to use 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.

Proposal:

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

Resolution: Rejected.

Heather's indication 20 Jul 2019: OK, makes sense to implement.

That would eliminate the need for the "bare" attribute mentioned in [Section 3.1.25](#).

This issue is tracked as [github issue #45](#)

3.1.10.2. 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.

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

Resolution: Rejected.

Heather's indication 20 Jul 2019: OK

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

This issue is tracked as [github issue #46](#)

3.1.11. 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?

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

Resolution: Accepted.

This issue is tracked as [github issue #47](#)

3.1.12. In Section 2.32, `<organization>`

3.1.12.1. Missing "asciiAbbrev" Attribute

The schema provides for extra attributes: "ascii" and "abbrev". Why no "asciiAbbrev" for the case when the name and abbreviation has non-ascii characters?

Proposal:

Add an attribute "asciiAbbrev" for <organization>, to provide abbreviated organization names in both ascii and non-ascii contexts.

Implementation: The current version of xml2rfc supports "asciiAbbrev".

Heather's indication 20 Jul 2019: OK

3.1.12.2. Attribute "showOnFrontPage"

Guidance from the IAB regarding IAB stream documents (<https://www.rfc-editor.org/materials/iab-format.txt>) indicates that "'Each author's name SHOULD be listed without an organization.". See also xml2rfc [ticket #311](#).

In [[RFC7991](#)] there is no way to turn on or off the display of <organization> on the front page, which would be needed for cases when it is not wanted IAB documents to show such on the front page. (Cases where display of <organization> is wanted is trivially supported by the current code).

In order to make it possible to expressly control this for a vocabulary version 3 XML document, version 2.21.0 of xml2rfc introduces an attribute "showOnFrontPage", with default value "true".

This issue is tracked as [github issue #36](#)

Heather's indication 20 Jul 2019: OK

3.1.13. In Section 2.37, <postal>

The enhancement to <postal>, adding a <postalLine> element, is a fair step on the way to permitting better representation of the wealth of postal addresses around the globe which don't match the American postal addresses.

Unfortunately, it manages to throw the baby out with the bathwater by constraining postalLine to be used only if none of the other elements are used. This makes it impossible to apply [hCard](#) [[HCARD](#)] labels (based on vCard [[RFC6350](#)] properties) to the elements of an address, as [[RFC7992](#)] requires. Applying the schema from [[RFC7991](#)] would make country information and hCard tags unavailable for any locality with a postal address scheme that needs to use <postalLine> because it does not match the American scheme. This would make statistics such as the author origin statistics either miss authors with such addresses, or make the statistics harder to compile than

is necessary, and make for instance the data on this page skewed:
<https://datatracker.ietf.org/stats/document/yearly/continent/>

The current implementation maps <postalLine> to the hCard property "extended-address", and permits it to be used together with other elements, in particular <country>, <region>, and <city>. This is a change to the schema.

The current implementation also provides a full set of hCard- and [RFC6350]-compatible address elements, including <extaddr> and <pobox>. The hCard locality address component is mapped to the current <city> element, however; not renamed to '<locality>'.

3.1.14. 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.

Proposal: 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.

Heather's indication 20 Jul 2019: OK

This issue is tracked as [github issue #48](#)

3.1.15. In Section 2.41, <referencegroup>

If <referencegroup> is to be used to represent for instance an STD entries that consist of multiple RFCs, the STD itself will have an URL. It would be natural to represent that with a "target" attribute, as for <reference>.

Proposal: Add a "target" attribute for <referencegroup>, matching the one for <reference>.

Implementation: Implemented in xml2rfc v 2.18.0

Heather's indication 20 Jul 2019: OK

3.1.16. 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.

Proposal: 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.

Resolution: Accepted.

Heather's indication 20 Jul 2019: OK

This issue is tracked as [github issue #49](#)

3.1.17. 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.

Proposal: 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.

Heather's indication 20 Jul 2019: OK

3.1.18. 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 4.4.25](#).

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

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

Heather's indication 20 Jul 2019: OK

3.1.19. 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.

Proposal: 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.

Heather's indication 20 Jul 2019: OK

3.1.20. In Section 2.46.2, "numbered" Attribute

The text indicates that only top-level sections may have numbered="false", and that a section with numbered="false" may not have a child section with numbered="true". But that leaves no value that is valid for child sections of an unnumbered section: They cannot have numbered="false", since they are not top-level sections, and they cannot have numbered="true", since the parent has numbered="false".

Additionally, the prohibition against child sections having numbered="false" removes the option of truncating the ToC listing for some child sections; without providing a good explanation for this limitation, it seems arbitrary and counter-intuitive to disallow this feature.

Proposal: Permit sections which are not top-level sections to have numbered="false".

Implementation: In The current version of xml2rfc, child sections may have numbered="false".

Heather's indication 20 Jul 2019: OK

3.1.21. In Section 2.47, <seriesInfo>

3.1.21.1. Too many possible combinations

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. New usages, with the purpose of replacing various attributes on the <rfc> element, have been added without any consistent pattern or table of permitted and forbidden combinations of values and attributes.

3.1.21.2. The "name" Attribute

The 'name' attribute is mandatory, and only 3 values are permitted: "RFC", "Interned-Draft", and "DOI", according to RFC 7991. 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., in the current reference library have have become illegal.

3.1.21.3. Incompatibility between v2 and v3 schema

The placement of <seriesInfo> elements within <reference> has changed in the v3 schema, in that it has been pulled into <front>, and the v2 placement has been deprecated. But this makes 'bibxml' reference files produced according to the v3 schema incompatible with v2 processors, and would require us to maintain 2 separate quotation libraries.

3.1.21.4. Inappropriate Introduction of the "stream" Attribute

The v3 specification in [[RFC7991](#)] introduces two new attributes with semantic content, in addition to the ASCII versions of the pre-existing "name" and "value" attributes: "stream" and "status".

The intention seems to be to deprecate attributes on <rfc>. However, these attributes cannot have multiple values for a document, which makes the move to <seriesInfo>, which can occur multiple times, dubious.

3.1.21.5. Summary

The number of issues introduced with the move of the <seriesInfo> element and its re-purposing in order to fill functionality in the

front of a document is wholly disproportionate with any added functionality. The specification [[RFC7991](#)] does not provide any rationale for the changes, and there seems to be no major benefits to the new schema.

Proposal: Do a rewrite of this that does not add new details to the already complex <seriesInfo> semantics, compared to the v2 vocabulary, 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/>.

The easiest approach would be to simply revert to the v2 semantics and placement of <seriesInfo> elements, with documentation of that.

Implementation: The current implementation does not strip or disregard the attributes on <rfc>; apart from that the schema is not reverted to v2 in the current implementation, but see also [Section 3.1.17](#), [Section 3.1.19](#) and [Section 3.2.2](#).

Heather's indication 20 Jul 2019: Starred, rewrite needed in order to simplify and clean this up.

3.1.22. In Section 2.48, <sourcecode>

The specification is not clear on emitting <CODE BEGINS> and <CODE ENDS> automatically when rendering <sourcecode>. In some cases it would be helpful, in others not.

Proposal: Add an attribute 'markers' for <sourcecode>, to control the emission of <CODE BEGINS> and <CODE ENDS>. If markers="true" and the "name" attribute is set, the filename will also be emitted, as specified in [[RFC8407](#)] for YANG modules.

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

Heather's indication 20 Jul 2019: OK

3.1.23. In Section 2.53.3 and 2.53.4.

3.1.23.1. Unnecessary limitation on the use of "keepWithNext"

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, or combinations thereof?

Proposal:

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

Implementation: Not in the current version of xml2rfc.

Heather's indication 25 Jul 2019: OK to implement.

3.1.23.2. 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](#)].

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

Implementation: Not in the current version of xml2rfc.

Heather's indication 20 Jul 2019: Undecided

3.1.24. 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. Such an expansion is already mandated by [Section 3.4](#) of [[RFC7997](#)]; but without schema support, it would be very hard for tools to enforce this. The text in [Appendix A.1](#) is proposed for inclusion in RFC 7991-bis as a new section.

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. Such an expansion is already mandated by [Section 3.4](#) of [\[RFC7997\]](#); but without schema support, it would be very hard for tools to enforce this. The text in [Appendix A.1](#) is proposed for inclusion in RFC 7991-bis as a new section.

Implementation: Implemented as described in [Appendix A.1](#).

Heather's indication 20 Jul 2019: Isn't this already required by 7997??

3.1.25. In Section 2.63.2, "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.

Proposal: 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 3.1.10](#) with a whitespace bullet could be used.

Heather's indication 25 Jul 2019: OK

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.

Heather's indication 25 Jul 2019: Questionable (see [Section 3.1.10.1](#) for alternative approach).

3.1.26. In Section 2.66.1, <xref> "format" attribute

3.1.26.1. The "derivedContent" attribute

For items in an ordered list, the "derivedContent" attribute should be set to the counter value for the item. But that counter value is only known during rendering. How is this supposed to work?

Proposal: In order to be able to set the "derivedContent" value, the preptool actually has to work through the list and derive the rendered counter. If we accept this, [\[KISS\]](#) and [\[DRY\]](#) both points in the direction of not discarding this value, but making a record of it, in the same manner as we make a record of "derivedContent" for <xref>. To do this, add a "derivedCounter" for , and fill it in with the calculated counter value.

Implementation: Implemented as proposed.

Heather's indication 25 Jul 2019: OK

3.1.26.2. Referencing a <dl> entry

It is specified that <xref> with format="counter" may reference sections, figures, tables, or ordered lists; but there does not seem to be any technical reason why this should not also be permitted for definition lists.

Proposal: Permit <xref> with format="counter" to also reference entries in definition list entries.

Implementation: Implemented as proposed.

Heather's indication 25 Jul 2019: OK

3.1.26.3. Combined effects of <xref> text, and the "format" and "sectionFormat" attributes.

If the <relref> functionality is folded into <xref> we are left with two format attributes, "format" and "sectionFormat". We then need to clearly specify if and how they interact. The following approach is suggested:

The "format" attribute should have effect only on the content of the internal link to the cited <reference> entry. If the "sectionFormat" attribute has a value of "bare", which does not cause any internal link to be rendered, the "format" attribute has no effect (or, possibly, is disallowed).

The "sectionFormat" attribute should have effect only on the rendering of the external link part. There is no "derivedSection"

attribute to match the "derivedFormat" attribute, the "section" attribute value is used in combination with the "sectionFormat" value when rendering the external link.

If an <xref> element with a "section" attribute value has text content, the text content is only used in the rendering of the internal link to the cited <reference>, with one exception: If the "sectionFormat" attribute value is "bare", then the <xref> text content is used to render a second external link in parentheses, after the initial external link that shows the external section number.

3.1.27. In Section 3.3, <format>

The [[RFC7991](#)] text seems to be based on a misunderstanding of the purpose of the <format> element in pointing to alternative representations of a reference. There seems to be no reason in removing this ability. The current implementation does not remove alternative <format> entries when converting v2 to v3. The RFC 7991 text should be adjusted accordingly, and in RFC 7992 it should be specified how to render links to alternative formats for a reference.

Heather's indication on 25 Jul 2019: Detailed proposal needed.

3.1.28. 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.

Proposal: Provide an attribute "indent" on <dl> as suggested in [Section 3.1.7](#).

Implementation: Implemented as proposed.

Heather's indication 25 Jul 2019: OK

3.1.29. 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).

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

Implementation: Done in the current version of xml2rfc.

Heather's indication 25 Jul 2019: OK

3.1.30. Use of the term "counter".

The classical meaning of this term is 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.

Proposal: 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.

Heather's indication 25 Jul 2019: Isn't "number" used for something else?

3.1.31. In Section 2.44, <relref>

(This section is out of order so as to not change the section numbering of previous sections while work is ongoing on a rfc7991bis document.)

The <relref> element has functionality that extends <xref>, and at first sight it is hard to distinguish the two. It would be better to remove <relref> and just add section, relative, and displayFormat to <xref>. Maybe change displayFormat to the earlier proposed 'sectionFormat'. (This point is also made in [Section 4.4.22](#))

Proposal: Deprecate <relref>.

Implementation: The current version of xml2rfc converts any occurrences of <relref> to the equivalent <xref> element.

Additional warnings about <relref> being deprecated would be in order.

3.1.32. In Section 2.66, <xref>

(This section is out of order so as to not change the section numbering of previous sections while work is ongoing on a rfc7991bis document.)

The <xref> element permits only plain text content, which limits how it can be used with explicit text. Permitting also , , <sub>, <sup>, and <tt> would make it possible to use the typographic expressions permitted in otherwise in running text. It also makes it possible for <toc> entries to reflect these typographic elements.

Proposal: Permit , , <sub>, <sup>, and <tt> to be used as children of <xref>

Implementation: Implemented in the current version of xml2rfc.

3.1.33. Contributor names

(This section is out of order so as to not change the section numbering of previous sections while work is ongoing on a rfc7991bis document.)

One thing that has been repeatedly requested both by the RPC and by RFC authors is a way to include contributor information in documents, also when the contributor names contain non-ASCII characters. This is applicable also for mention of names and possibly contact details of other persons than contributors and authors (even if the contributor case is the one that comes up most often).

Proposal: Permit a <contact> element to be used to provide name and address information for persons that aren't authors. This will allow both non-ASCII and ASCII-equivalence name information to be provided and rendered, in much the same way that author information is rendered.

Implementation: Implemented in the current version of xml2rfc. The <contact> element is allowed in two contexts: As a direct child of <section>, where it will be rendered in the same manner that author information is rendered in the Authors' Addresses section, and as a child of <t>, where it will be rendered inline, in a similar manner that author information is rendered in citations.

3.2. RFC 7998

3.2.1. New Section 5.1.6, Attribute validation

Some attribute validation beyond what the schema enforces is possible and desirable. One example of this is to validate that all attributes which are expected to have integer values actually does so. A section on this should be added. The current implementation adds integer attribute validation and verification that apart from the name attributes of <author>, no attribute values have non-ASCII content.

Heather's indication 25 Jul 2019: Good idea!

3.2.2. 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.

Proposal: 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 of <seriesInfo> from the schema. This is not a problem from a rendering perspective, since the "stream" attribute does not need to have a value in order for the <seriesInfo> to be rendered correctly (most instances of <seriesInfo> in the current bibxml library does not have a "stream" attribute set).

3.2.3. 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 are <dt> and present, but not , <dl>, ?

Proposal: 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>, , , which all are allowed to have pn= attributes according to the schema.

Heather's indication 25 Jul 2019:

OK

3.3. Some attributes should have value type xsd:ID

In generated HTML, the values set for "pn" and "slugifiedName" will be used as link targets, which makes a type of xsd:ID appropriate in the input format, as this will guarantee that they all have distinct values in the xml source.

Proposal: Change the "pn" and "slugifiedName" to type xsd:ID.

Implementation: Implemented in the current version of xml2rfc.

Heather's indication 25 Jul 2019: OK

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 "type" is missing.

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.

Proposal: 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.

Heather's indication 25 Jul 2019: OK

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.

Heather's indication 25 Jul 2019: TODO

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.

Proposal: 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.

Heather's indication 25 Jul 2019: TODO

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.

Proposal: 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](#)].

Heather's indication 25 Jul 2019: OK

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.

Proposal: 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 permits the <date> element to have text content, as an alternative to year, month, and day attributes.

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.

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

Heather's indication 25 Jul 2019: OK

4.1.5. In Section 2.48.4, "type" 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.

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

Heather's indication 25 Jul 2019: OK

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.

Proposal: 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.

Heather's indication 25 Jul 2019: OK

4.1.7. In Appendix B.2.1: Generation of PN numbers

The current specification says:

- "pn" for all elements not listed above always has the format "p-nnn-mmm", where "nnn" is the section number and "mmm" is the relative position in the section. For example, this would be "p-2.1.3-7" for the seventh part number in Section 2.1.3.

However, this will result in counting up the part numbers for invisible parts, when numbered elements are contained within enclosing numbered block elements.

The current implementation instead uses the same "pn" numbering scheme as Julian Reschke's vocabulary v3 XSLT processor, where both the section number part and the relative position within the section has hierarchical numbering. For instance, the second element in Section 2.1 would have a pn number of "section-2.1-2", and assuming it is a dl element, the first dt element within the dl in Section 2.1 would have a pn number of "section-2.1-2.1".

4.2. RFC 7992

4.2.1. In Section 5.1, IDs

The current specification says:

HTML elements that are generated from XML elements that include an "anchor" attribute will use the value of the "anchor" attribute as the value of the "id" attribute of the corresponding HTML element. The prep tool produces XML with "anchor" attributes in all elements that need them. Some HTML constructs (such as <section>) will use multiple instances of these identifiers.

But I believe HTML5 does not permit more than one "id" attribute per element, which begs the question of how <section> will use multiple instances of identifiers?

4.2.2. In Section 6.2, Root Element

Typo:

OLD: <seriesInfo> element's "name" attributes

NEW: <seriesInfo> elements' "name" attributes

4.2.3. In Section 6.4, Page Headers and Footers

This is incomplete. It gives an example, but does not specify how it is to be filled in.

Is the formatter expected to fill out the cells, based on the pattern given, or is that supposed to happen magically based on WD-css3-page-20130314 ?

If the cell content is supposed to be provided by the formatter, it would be good to have a bit more specification than the example; if not, it would be nice for that to be stated explicitly.

The mention of the '[Page]' placeholder could be taken as an indication that all cell content shown are placeholders, but are they, really?

Implementation: The current implementation has code to insert placeholder HTML, but not code to fill in the cells with actual information from the document. Since this is meaningless if the guess is wrong, this code has been disabled for now.

The current implementation insteads adds CSS that explicitly sets the header and footer text to the desired values.

Heather's indication 25 Jul 2019: Rewrite needed

4.2.4. In Section 6.5, Document Information

This information seems to be scrambled and incomplete. It suggests the use of 'Status:' for what is otherwise called 'Category:'. It simplifies the presentation of series information to the point that no clue is given of how to handle the two bits of information related to series name and series number -- the example shows 'Series:' 'Internet-Draft', which gives no guidance at all. There is no mention of whether to display 'Obsoletes:' and 'Updates:' information or not.

On a more general note, this is the second section where an incomplete example is provided instead of specification. Examples are however not replacements for proper specification; they are at best a help in making a specification real to the user. Both this section and [Section 4.2.3](#) needs to be expanded to provide a complete specification.

Styling query: The example gives the style of the element that holds author initials the class 'initial' while the attribute is appropriately named 'initials'. Is the difference in attribute and style names intentional? In any case, 'initials' would be more appropriate.

Implementation: Instead of trying to follow what's written, the current implementation tries to provide the same fields and information which is provided by the text/plain formatter, in a sensible way. This is guesswork.

The implementation also has used the sample HTML document for guidance here, in order to be able to progress with something that works with the style sheet from the RFC-Format CSS project.

Heather's indication 25 Jul 2019: Rewrite needed

4.2.5. 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?

4.2.6. Inconsistent use of "s-", "n-" and User-Supplied "id" Attributes

RFC 7991 [[RFC7991](#)] specifies an attribute "slugifiedName" on <name>, but does not specify how it is to be used. RFC 7998 [[RFC7998](#)] specifies how to create these, but not how they should be used. In RFC 7992, slugified names, with an "n-" (or "name-") prefix, are sometimes used on sections, sometimes not. "s-" (or "section-") IDs are sometimes used on <h2> and other header elements, sometimes on paragraph, divs, asides, blockquotes etc. Section 9.33 of [[RFC7992](#)] even uses a reference to an "n-" ID that doesn't exist, although it clearly should, based on the section name. This is a mess.

Implementation: The implementation consistently transfers the "slugifiedName" attribute on <name> to an "id" attribute on the <h2> or other header element generated from the name. Section numbers ("s-" or "section-" values) from "pn" attributes are consistently transferred to the <section>, <p> or other HTML element generated from the XML element on which they appear. User-supplied "anchor" attributes on XML elements are consistently transferred to a <div> inside the HTML element generated from the XML element with the anchor, encapsulating the content generated from the XML element.

4.2.7. In Section 9.2, <address>

The example reiterates an abbreviated form of the xml given under <author>, as if there was no difference between the rendering of <address> and <author>. Furthermore, the example shows only rendering of elements which are *not* part of <address>; any rendering of the elements contained within <address> is omitted. This is misleading, in particular since rendering of the individual child elements (<postal>, <phone> , <facsimile> , <email>, and <uri>) *has* been specified to have explicit renderings.

Given that the specification text is reasonable for author name and org, but nonsense for the <address> element, the following text has been assumed during implementation:

The <address> element will be rendered as a sequence of <div> elements, each corresponding to a child element of <address>, and enclosed in the same <address> element as the name, role, and

organization information. Element classes will be taken from hCard [HCARD], as specified on <http://microformats.org/wiki/hcard>.

This is the mapping used by xml2rfc from the address fields to hCard properties:

i18n address field	xml2rfc element	hCard property
-	<extaddr>	extended-address
street_address	<street>	street-address
sorting_code	<sortingcode>	postal-code
postal_code	<code>	postal-code
city_area	<cityarea>	locality
city	<city>	locality
country_area	<region>	country-area
country_name	<country>	country-name
-	<pobox>	post-office-box

Table 1

4.2.8. In Section 9.7.2, Authors of this Document

RFC 7997 gives the text separating the ASCII and non-ASCII address information as "Additional contact information:".

RFC 7997 manages to convey the desired rendering order of ASCII and non-ASCII address information without any americentric language, but RFC 7992 talks about the non-ASCII version as 'fallback'. As a non-native English speaker raised speaking and writing 2 languages that both have alphabets with non-ASCII letters, the author of this memo finds the language in RFC 7992 somewhat offensive, and suggests that it be removed from the document.

The current xml2rfc implementation uses the layout and wording given in RFC 7997, not RFC 7992.

Furthermore, the document also says:

"When the <author> element, or any of its descendant elements, has any attribute that starts with "ascii", all of the author information is displayed twice. ..."

This is in conflict with [RFC7997], Section 3.2, which indicates that the determining factor for displaying both non-ASCII and ASCII author information is whether a script outside the Unicode Latin blocks is used for the primary information. The current implementation checks for this, rather than going by the presence of attributes with an 'ascii' prefix.

4.2.9. In Section 9.7.3, Authors of References

Information is completely missing on how to render non-ascii name information in references.

4.2.10. In Section 9.16, <cref>

The text does not mention how to deal with <cref>s with `display="false"`. Presumably by not displaying them; but if there exists internal links to the <cref> anchor, completely omitting the rendering could cause breakage. The current xml2rfc implementation handles this by inserting an empty HTML with the appropriate id attribute.

4.2.11. In Section 9.24, <eref>

No handling is provided for the case where the <eref> element is empty, which would result in an empty (and invisible) HTML <a> element. The current implementation in this case instead inserts a span containing '<', an <a> with appropriate href and the target URL as text, and '>'.

4.2.12. In Section 9.25, <figure>

The specified HTML rendering will result in a figure title text which links to itself. With the caption placed below the figure, this means that if you click on the title, the figure will scroll up above the browser window. This is not particularly useful.

The current implementation instead inserts an empty as the first element of the figure, and gives it an id attribute with the value set to the slugifiedName attribute of the <name> element, in order to make the link from the figure caption text useful.

4.2.13. In Section 9.27, <iref>

The text refers to the "irefid" attribute. Interpreted as meaning the "pn" attribute, as the schema has no "irefid" attribute.

4.2.14. In Section 9.33, <note>

Typo: `s/"yes"/"true"/`

**4.2.15. In Section 9.34, **

The element has no "style" attribute. The implementation assumes "type" instead.

4.2.16. In Section 9.35, <organization>

The text here is in conflict with RFC 7997 with respect to rendering the Authors' Addresses section. RFC 7997 describes rendering two sets of full information, one ASCII and one non-ASCII, not a single <div> where the non-ASCII name is given first, followed by the ASCII version as needed.

4.2.17. In Section 9.36, <phone>

The text here is in conflict with the use of 'type' in vCard and hCard. Telephone number type annotations identify things like 'Home' and 'Work'. The current implementation does not add the uppercase VOICE type annotation.

4.2.18. In Section 9.37, <postal>

The current specification says:

This element renders as an HTML <div> with CSS class "adr", unless it contains one or more <postalLine> child elements; in which case, it renders as an HTML <pre> element with CSS class "label".

Handling <postalLine> elements this way violates the [hCard](#) [HCARD] specification. They will instead be rendered as hCard elements with class "extended-address" within the same <div> with CSS class "adr" as other <postal> sub-elements.

The specification continues to enforce American postal address structure on addresses that don't use <postalLine>. This has been changed in the current implementation; instead of using the fixed American layout for all countries, the formatting has been adapted to use country-specific formatting for all recognised country names and codes.

(The implementer considered applying a non-US postal address layout for all US addresses, to see how swiftly this would raise hue and cry and be labelled a bug, but in the interest of not causing unnecessary upset resisted the urge.)

4.2.19. In Section 9.40, <reference>

4.2.19.1. Misleading example

Section 9.41 of [\[RFC7992\]](#) shows <referencegroup> being rendered as <dt>, <dd>, while the example for this section shows one reference being rendered as <dl> <dt> <dd> </dl>. This is contradictory. Which one is right? The CSS class on <dl>, which is specified as class="reference" points in the direction that each individual

<reference> entry should be rendered as one <dl> with one set of <dt> <dd>, while it would seem much more logical to render the list of references as one single list holding all the references.

The current xml2rfc implementation renders <references> as a section containing one <dl>, and each individual <reference> or <referencegroup> as a <dt> <dd> pair within that list. To match this, the CSS class used is 'references' rather than 'reference'.

4.2.19.2. Anchor handling disregards <displayreference>

There is no mention in the description of the HTML rendering of <reference> of the effects of <displayreference>, which definitely needs to be considered. Emitting the original anchor value from the reference entry (which often comes from the bibxml reference library) would make the emitted reference labels wrong when there is a <displayreference> entry for the reference. The most straightforward approach would be to add an attribute "derivedAnchor" to <reference> and have the preptool set it.

Proposal: Add an attribute "derivedAnchor" to <reference>. Specify in [RFC7998] that this is set by the preptool, and update [RFC7991] and [RFC7992] accordingly.

Implementation: Implemented as proposed.

4.2.19.3. Handling of author lists in <reference> is under-specified

The example shows the 'and' between author names within a span (unclear why) but does not show how to handle commas separating authors. The style examples on github do not enclose commas or 'and' in a span, which seems reasonable. Going with the style example files here. Section 9.7.3 of RFC 7992 gives an example without 'and' enclosed in a span, contradicting Section 9.40 of the same RFC.

Trying to sort out the rendering of author names in references by looking at other sources than RFC 7992 reveals that the CSS samples show dual reference entries, one with ascii names and another with non-ascii names. This contradicts RFC 7997, which shows a single reference entry where the non-ascii author names are given with the ascii equivalent in parentheses.

The current implementation follows RFC 7997 in this respect, not RFC 7992.

4.2.20. In Section 9.41, <referencegroup>

This element is a sibling to <reference>, and <reference> is described as being rendered as a <dl> with one set of <dt>, <dd> child elements.

However, `<referencegroup>` is specified to be rendered as a `<dt>`, `<dd>` set, **without** any containing `<dl>`. The individual reference entries are then specified to be rendered as `<div>`s inside the `<dd>`

1. This produces invalid HTML, because there is no containing `<dl>`
2. Why isn't this rendered as a `<dl>` with multiple `<dd>` entries?
That would make the styling much more consistent.

4.2.21. In Section 9.42, `<references>`

The specification says that this is to be rendered as a `<section>`. However, if `<reference>`s and `<referencegroup>`s are to be rendered as `<dt>`, `<dd>`, then this element needs to be rendered as `<section> <dl> ... </dl> </section>`

4.2.22. In Section 9.54, `<table>`

RFC 7992 says: "This element is directly rendered as its HTML counterpart."

This ignores the special processing needed to insert a `<caption>` element. The current implementation handles this appropriately. The specification should be updated.

4.2.23. In Section 9.56, `<td>`

RFC 7992 says: "This element is directly rendered as its HTML counterpart."

However, that is not correct. An appropriate style class needs to be inserted to honour the "align" attribute. The classes "alignLeft", "alignCenter", and "alignRight" of the provided CSS are geared towards block alignment; here text alignment is needed. The current implementation uses "text-left", "text-center", and "text-right", and provides appropriate CSS entries. (These attribute names matches the equivalent bootstrap names.)

4.2.24. In Section 9.58, `<th>`

RFC 7992 says: "This element is directly rendered as its HTML counterpart."

However, that is not correct. An appropriate style class needs to be inserted to honour the "align" attribute. The classes "alignLeft", "alignCenter", and "alignRight" of the provided CSS are geared towards block alignment; here text alignment is needed. The current implementation uses "text-left", "text-center", and "text-right", and provides appropriate CSS entries. (These attribute names matches the equivalent bootstrap names.)

4.2.25. In Section 9.60, <title>

This section completely lacks specification on how to render title elements with non-Latin content and an "ascii" attribute.

4.2.26. In Section 9.66, <xref>

The specification says:

... If the "format" attribute has the value "default", and the "target" attribute points to a <reference> or <referencegroup> element, then the generated <a> element is surrounded by square brackets in the output.

However, inspection of actual usage indicates that a better rendering would be to surround the generated <a> with square brackets only for empty <xref> elements; when there is content, usage indicates that authors provide enclosing parentheses or not depending on circumstances. Since in HTML rendering the brackets are not necessary to provide a clue that this refers to other content (unlike the text case), the square brackets could be omitted when the <xref> element contains text. The current implementation does so.

4.2.27. In Section 9.18, <dd>

(This section is out of order so as to not change the section numbering of previous sections while work is ongoing on a rfc7991bis document.)

The text does not mention pilcrow insertion. Having pilcrow on other list items, but not on this one turns out to be surprising to users. Applying the same text about pilcrow insertion as for other list entries seems indicated: "If there is no contained element that has a pilcrow (Section 5.2) attached, a pilcrow is added."

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.

Proposal: 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?

Proposal: 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.

Proposal: 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.

Proposal: 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.

Proposal: 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.

Proposal: 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).

Proposal: 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'.

Proposal: 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?

Proposal: 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.

Proposal: 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

Proposal: 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 3.1.23.2](#) 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.

Proposal: 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.

Proposal: 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 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 according to [\[DRY\]](#), and unnecessarily introduces the possibility of conflict, as a result of multiple <seriesInfo> elements being permitted (Relevant to the v3 schema, not the preptool).

Proposal: 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.

Proposal:

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.

Proposal: 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.

Proposal: 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.

Proposal: 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.

Proposal: 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.

Proposal:

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.

Proposal: 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 5.4.8.1, "derivedContent" Insertion (with Content)

This section is problematic. It says:

For each <xref> element that has content, fill the "derivedContent" with the element content, having first trimmed the whitespace from ends of content text. Issue a warning if the

"derivedContent" attribute already exists and has a different value from what was being filled in.

On the surface, it seems to replace the effect of using `<xref>` with `format="none"` under vocabulary version 2, but in practice it blocks the combination of generated text (say a section number fetched from the referenced section) with author-provided text, since any author-provided text will preempt generated text that is based on the "format" attribute with the author-provided text.

Additionally, and in one sense just as bad, it violates the principle of least surprise [[POLA](#)], since it is a fundamental change from how text inside the `<xref>` element was combined with generated text in version 2.

Implementation: As of xml2rfc 2.19.0, the expansion of `<xref>` and its variation based on "format" attribute settings has been reverted to be more in line with version two, and more regular. The attribute setting `format="none"` is honoured again, and if the `<xref>` element has text content, it is combined with the content derived from the format attribute setting, rather than simply overriding it, as was the consequence of Section 5.4.8.1 of [[RFC7998](#)].

- *Derived content is generated based on the format attribute
- *If text content is provided, it is shown together with any derived content
- *If the `<xref>` target is a listed reference, the derived content is shown within square brackets
- *If the `<xref>` target is not a listed reference , the derived content is shown within parentheses if there is text content, and without parentheses if not.
- *If text content is provided, and is identical with the derived content, it is ignored.

This addresses [github issue #17](#).

4.4.18. 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.

4.4.19. 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.

Proposal: 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.

4.4.20. 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.

Proposal: 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.

4.4.21. 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.

Proposal: 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.

4.4.22. 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'.

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

Implementation: The <relref> functionality has been folded into <xref>. As of version 2.20.0, xml2rfc rewrites <relref> to <xref>, with "displayFormat" changed to "sectionFormat".

4.4.23. New Section 5.4.10, Unused Reference Warnings

During vocabulary version 2 processing, warnings are emitted for <reference> entries that are not used. This is not specified for v3, but is desired, according to RFC Editor staff. Implemented in xml2rfc v2.18.0.

4.4.24. New Section 5.4.11, 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."

Proposal:

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

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

4.4.25. In Section 5.6.3, <link> Processing

4.4.25.1. Using docName to generate convertedFrom

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.

Proposal: 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 3.1.18](#)).

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

4.4.25.2. Invalid "rel" values.

Using the W3C validator to validate the V3 output, there are some errors; all of them with the same basic complaint: The values prescribed for the <link> "rel" attribute (derivedFrom, describedBy, and item) are not permitted values. The permitted values are given here: <https://html.spec.whatwg.org/multipage/links.html#linkTypes>.

5. Possible New Work

5.1. Inline and Display Math

Various people have repeatedly asked for better provisions for using math in drafts. A number of different cases seems to be mentioned, listed roughly in order of complexity:

1. Ability to use individual math symbols in running text
2. Ability to insert math equations in running text
3. Ability to display complex math as figures

Not surprisingly, these 3 use cases correspond quite well to the 3 modes the TeX typesetting system uses when considering math. In text mode, individual math symbols may be inserted in running text; in inline math mode, equations can be built, but will be displayed

inline with preceding and following text (but can still display content that does not otherwise fit on a single line, such as a fraction, to take a very simple example. In display math mode, equations will be displayed on a separate line (or multiple lines).

One possible approach to integrating the second and third case above in IETF documents rendered as HTML and PDF is to add support for [MathJax](#) [[MATH-JAX](#)].

For the first case, the simplest and most straightforward approach would be to extend the permitted range of unicode code-points in running body text from permitting ASCII only (unless enclosed in `<u>` elements) to permit ASCII and code-points in the 'Mathematical Operators' and 'Supplementary Mathematical Operators' blocks, or alternatively code-points in the 'Symbol, Math' ('Sm') category, which includes the mentioned blocks, but also includes additional symbols, and possibly goes further than needed if the second case above is provided for.

5.2. Change Bars

Change bars have been used in published RFCs (see for instance [[RFC6818](#)]), and would seem to be a fairly obvious extension to add. When documents were produced with a possible nroff step, adding change bars was fairly straightforward, but with the transition to publication in XML as the archival format, this capability has now been lost, unless we introduce an element that provides it.

5.3. Element Nesting

There exists a number of areas where without any obvious reason some elements have been excluded from appearing. For instance, in Section 3.1 of RFC-to-be 8646 (draft-ietf-manet-dlep-pause-extension), there is a small table as part of one `<dd>` in a `<dl>` (definition list) that defines the bit fields of a data item. But since `<table>` is not permitted in a `<dd>`, the definition list has to be broken at this point, a `<table>` element inserted, and then the `<dl>` continues to define the rest of the data item fields.

This seems quite unnecessary; there is no obvious reason why tables cannot be part of `<dd>` or `<li>`.

5.4. Schema Consistency

After some changes introduced during implementation, such as permitting `<blockquote>` within `<li>` after [discussion on the xml2rfc-dev list](#) there is a lack of consistency in which elements are permitted where. Once a decision has been made on the additional proposed changes above, (such as permitting `<table>` within `<dd>` and

) a review should be done of the resulting schema, and see if a cleanup for consistency is needed.

Here are for instance lists of elements permitted inside various block elements, as of release 2.28.0 of xml2rfc:

aside:

(artset| artwork| dl| figure| iref| ol| t| table| ul)*

blockquote:

(artset| artwork| dl| figure| ol| sourcecode| t| ul)+

li: (artset| artwork| blockquote| dl| figure| ol| sourcecode| t| ul)+

dd: (artset| artwork| dl| figure| ol| sourcecode| t| ul)+

td: (artset| artwork| dl| figure| ol| sourcecode| t| ul)+

th: (artset| artwork| dl| figure| ol| sourcecode| t| ul)+

6. Security Considerations

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

7. Informative References

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

[HCARD] Celik, T., "hCard 1.0", 2015, <<http://microformats.org/wiki/hcard>>.

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

[MATH-JAX] The MathJax Team, "MathJax - Beautiful math in all browsers", 2009-2019, <<https://www.mathjax.org>>.

[POLA] Wikipedia, "Principle of least astonishment", 2018, <https://en.wikipedia.org/wiki/Principle_of_least_astonishment>.

[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>>.

[RFC6350]

Perreault, S., "vCard Format Specification", RFC 6350, DOI 10.17487/RFC6350, August 2011, <<https://www.rfc-editor.org/info/rfc6350>>.

[RFC6818]

Yee, P., "Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile", RFC 6818, DOI 10.17487/RFC6818, January 2013, <<https://www.rfc-editor.org/info/rfc6818>>.

[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>>.

[RFC8407]

Bierman, A., "Guidelines for Authors and Reviewers of Documents Containing YANG Data Models", BCP 216, RFC

8407, DOI 10.17487/RFC8407, October 2018, <<https://www.rfc-editor.org/info/rfc8407>>.

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

Appendix A. Proposed new sections in RFC 7991 bis

A.1. <u>

In xml2rfc vocabulary version 3, the elements <author>, <organisation>, <street>, <city>, <region>, <code>, <country>, <postalline>, <email>, <seriesInfo>, and <title> 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.

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 a "format" attribute. This provides a generalised means of generating the 6 methods of Unicode renderings listed in [RFC7997], Section 3.4, and also several others found in for instance the RFC Format Tools example rendering of RFC 7700, at <https://rfc-format.github.io/draft-iab-rfc-css-bis/sample2-v2.html>.

The "format" attribute accepts either a simplified format specification, or a full format string with placeholders for the various possible Unicode expansions.

A.1.1. Expansion of simplified <u> format specifications

The simplified format consists of dash-separated keywords, where each keyword represents a possible expansion of the Unicode character or string; use for example <u "lit-num-name">foo</u> to expand the text to its literal value, code point values, and code point names.

A combination of up to 3 of the following keywords may be used, separated by dashes: "num", "lit", "name", "ascii", "char". The keywords are expanded as follows and combined, with the second and third enclosed in parentheses (if present):

"num" The numeric value(s) of the element text, in U+1234 notation

"name" The Unicode name(s) of the element text

"lit" The literal element text, enclosed in quotes

"char"

The literal element text, without quotes

"ascii" The value of the 'ascii' attribute on the <u> element

In order to ensure that no specification mistakes can result for rendering methods that cannot render all Unicode code points, "num" **MUST** always be part of the specified format.

The default value of the "format" attribute is "lit-name-num".

A.1.1.1. Examples

Examples:

format="num-lit":

Temperature changes in the Temperature Control Protocol are indicated by the character U+0394 ("Δ").

format="num-name":

Temperature changes in the Temperature Control Protocol are indicated by the character U+0394 (GREEK CAPITAL LETTER DELTA).

format="num-lit-name":

Temperature changes in the Temperature Control Protocol are indicated by the character U+0394 ("Δ", GREEK CAPITAL LETTER DELTA).

format="num-name-lit":

Temperature changes in the Temperature Control Protocol are indicated by the character U+0394 (GREEK CAPITAL LETTER DELTA, "Δ").

format="name-lit-num":

Temperature changes in the Temperature Control Protocol are indicated by the character GREEK CAPITAL LETTER DELTA ("Δ", U+0394).

format="lit-name-num":

Temperature changes in the Temperature Control Protocol are indicated by the character "Δ" (GREEK CAPITAL LETTER DELTA, U+0394).

A.1.1.2. Expansion of <u> multi-codepoint strings

If the <u> element encloses a sequence of Unicode codepoints, rather than a single one, the rendering reflects this. The element

<u format="num-lit"> </u>

will be expanded to 'U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 (" ")'.

Unicode characters in document text which are not enclosed in <u> will be replaced with a question mark (?) and a warning will be issued.

A.1.2. Non-simplified <u> format specifications

In order to provide for cases where the simplified format above is insufficient, without relinquishing the requirement that the number of a code point always must be rendered, the "format" attribute can also accept a full format string. This format uses placeholders which consist of any of the key words above enclosed in curly braces; outside of this, any ascii text is permissible. For example,

The <u format="{lit} character ({num})">Δ</u>.

will be rendered as

The "Δ" character (U+0394).

As for the simplified format, "num" MUST always be part of the specified format in order to ensure that no specification mistakes can result for rendering methods that cannot render all Unicode code points,

A.1.3. Split expansion of <u> elements

There are cases which cannot be handled with either the simplified or full <u> format specifications. One is exemplified in Table 1 of the CSS sample document at <https://rfc-format.github.io/draft-iab-rfc-css-bis/sample2-v2.html#s-3>. Rendering this with <u> elements requires that the non-ascii content be rendered in one place (a table cell in one column) while the expansion is rendered in another cell in a different column. Provision for this has been made by modifying the expansion of <u> when it is referenced by an <xref>. This table, with <u> elements referenced by <xref> instances:

```

<table>
  <name>A Sample of Legal Nicknames</name>
  <thead>
    <tr>
      <th>#</th>
      <th>Nickname</th>
      <th>Output for comparison</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <td>1</td>
      <td>&lt;Foo&gt;</td>
      <td>&lt;foo&gt;</td>
    </tr>
    <tr>
      <td>2</td>
      <td>&lt;foo&gt;</td>
      <td>&lt;foo&gt;</td> </tr>
    <tr>
      <td>3</td>
      <td>&lt;Foo Bar&gt;</td>
      <td>&lt;foo bar&gt;</td>
    </tr>
    <tr>
      <td>4</td>
      <td>&lt;foo bar&gt;</td>
      <td>&lt;foo bar&gt;</td>
    </tr>
    <tr>
      <td>5</td>
      <td>
        &lt;
        <u format="name-num" anchor="greek-upper-sigma">Σ</u>
        &gt;
      </td>
      <td> <xref target="greek-upper-sigma" /> </td>
    </tr>
    <tr>
      <td>6</td>
      <td>
        &lt;
        <u format="name-num" anchor="greek-lower-sigma">σ</u>
        &gt;
      </td>
      <td> <xref target="greek-lower-sigma" /> </td>
    </tr>
    <tr>
      <td>7</td>

```

```

        <td>
            &lt;
            <u format="name-num" anchor="greek-final-sigma">ζ</u>
            &gt;
        </td>
        <td> <xref target="greek-final-sigma" /> </td>
    </tr>
    <tr>
        <td>8</td>
        <td>
            &lt;
            <u format="name-num" anchor="black-chess-king">♚</u>
            &gt;
        </td>
        <td> <xref target="black-chess-king" format="default"/> </td>
    </tr>
    <tr>
        <td>9</td>
        <td>
            &lt;Richard
            <u format="{char}> ({num})" anchor="richard-iv">IV</u>
            &gt;
        </td>
        <td>&lt;richard iv&gt;</td>
    </tr>
</tbody>
</table>

```

comes out as shown below:

#	Nickname	Output for comparison
1	<Foo>	<foo>
2	<foo>	<foo>
3	<Foo Bar>	<foo bar>
4	<foo bar>	<foo bar>
5	<Σ>	GREEK CAPITAL LETTER SIGMA (U+03A3)
6	<σ>	GREEK SMALL LETTER SIGMA (U+03C3)
7	<ς>	GREEK SMALL LETTER FINAL SIGMA (U+03C2)
8	<♠>	BLACK CHESS KING (U+265A)
9	<Richard IV> (U+2163)	<richard iv>

Table 2: A Sample of Legal Nicknames

A.2. <rendering>

The preferred ordering of postal address lines differs not only as a function of country and region, but also as a function of script for some countries (some locales will order addresses in native scripts from largest to smallest geographic areas, while romanised addresses are ordered from smallest to largest, for instance), and as a function of time (i.e., address formats and renderings can change over time).

The current implementation of <postal> rendering relies on an external library that provides correct ordering of native and romanised address formats as a function of country and region.

This does however not provide for stability over time -- if the address format changes over time, the rendering order indicated by the address internationalization library will reflect only what is appropriate at a particular point in time, which may be different from what was appropriate at the time a document was published.

In order to capture the appropriate address format for native and romanised postal addresses, at the time of publication, the <postal> element may have one or more (normally two) <rendering> child elements, according to the following Relax-NG Compact schema:

```

postal =
  element postal {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (( city | cityarea | code | country | extaddr | pobox | region
      | sortingcode | street)*
      | postalline+),
    rendering*
  }

rendering =
  element rendering {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute type { "ascii" | "unicode" },
    postalline+
  }

```

When <rendering> elements are present under <postal>, there should be one element with attribute type="ascii" if there is no non-Latin postal address content, and two elements, with types "ascii" and "unicode" respectively if there is non-Latin content present.

When producing prepped output, the preptool should invoke the address internationalization library to get the correct ordering of native and romanised address lines if the required <rendering> elements aren't already present, and insert <rendering> elements reflecting the ordering at the time of prepping. This will permit the capture of address line rendering order at the time of publication, and avoid re-invocation of the address internationalization library at a later time.

Renderers MUST use the appropriate <rendering> entries if they are present, rather than re-invoking an address internationalization library to get the address line ordering.

The expectation is that authors will provide the appropriate <postal> elements (<street>, <city>, <country> etc.), but not <rendering> elements; these will be filled in by the preptool and used by the renderers for all subsequent rendering.

Author's Address

Henrik Levkowetz
 Elf Tools AB
 Ollonstigen 8
 SE-18164 Lidingö
 Sweden

Email: henrik@levkowetz.com