Network Working Group
Internet-Draft
Obsoletes: 2629 (if approved)
Intended status: Informational
Expires: January 3, 2015

P. Hoffman
VPN Consortium

July 2, 2014

The 'XML2RFC' version 3 Vocabulary
draft-hoffman-xml2rfc-09

Abstract

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 2629 and its expected followup, draft-reschke-xml2rfc.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft takes place on the rfc-interest mailing list (rfc-interest@rfc-editor.org), which has its home page at <https://www.rfc-editor.org/mailman/listinfo/rfc-interest>.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes version 3 ('v3') of the 'XML2RFC' vocabulary; an XML-based language ('Extensible Markup Language', [XML]) used for writing RFCs ([RFCSTYLE]) and Internet-Drafts ([IDGUIDE]).

This document obsoletes the version ("v2") vocabulary [XML2RFCv2], which contains the extended language definition. That document in turn obsoletes the original version ("v1") [RFC2629]. This document directly copies the material from [XML2RFCv2] where possible; as that document makes its way toward RFC publication, this document will incorporate as many of the changes as possible. [[anchor2: More needs to be said here about what "obsoletes" means in this case.]]

The v3 format will be used as part of the new RFC series described in [RFC6949]. The new format will be handled by one or more new processors for checking the XML and converting it to other representations. Features of the expected processors are described in Appendix B.

Note that the vocabulary contains certain constructs that might not be used when generating of the final text; however, they can provide useful data for other uses (such index generation, populating a keyword database, or syntax checks).

In this document, the term "format" is used when describing types of documents, primarily XML and HTML. The term "representation" is used when talking about a specific instatiation of a format, such as an XML document or an HTML document that was created by an XML document.

The following two sections are a hopefully-complete list of all the technical changes between [XML2RFCv2] and this document, as well as the design criteria for those changes.

1.1. Design Criteria for the Changes in v3

The design criteria of the changes from v2 to v3 are:

- The intention is that starting and editing a v3 document will be easier than for a v2 document.

- There will be good v2-to-v3 conversion tools for when an author wants to change versions.

- There are no current plans to make v3 XML the required submission representation for drafts or RFCs. That might happen eventually, but it is likely to be years away.

There is a desire to keep as much of the v2 grammar as makes sense within the above design criteria and not to make gratuitous changes to the v2 grammar. Another way to say this is "we would rather encourage backward compatibility but not be constrained by it". Still, the goal of starting and editing a v3 document being easier than for a v2 document is more important than backwards compatibility with v2, given the latter two design criteria.

v3 is upwards compatible with v2, meaning that a v2 document is meant to be a valid v3 document as well. However, some features of v2 are deprecated in v3 in favor of new elements. Deprecated features are listed in Section 1.2.3, and are described in [XML2RFCv2].

1.2. Differences from v2 to v3

The format changes in v3 are listed in the following subsections.

1.2.1. New Elements in v3

- Add <dl>, <ul>, and <ol> as new ways to make lists. This is a significant change from v2 in that the child under these elements is <li>, not <t>. <li> has a model of either containing one or more <t> elements, or containing the flowing text normally found in <t>.

- Add <strong>, <b>, <em>, <i>, <tt>, <sub>, and <sup> for character formatting.

- Add <aside> for incidental text that will be indented when displayed.

- Add <sourcecode> to differentiate from <artwork>.

- Add <blockquote> to indicate a quotation as in a paragraph-like format.

- Add <titleelement> to sections, figures, and texttables to allow character formatting (fixed-width font) in their titles, and to allow references.

- Add <postalLine>, free text that represents one line of the address.

- Add <displayreference> to allow display of more mneumonic anchor names for automatically-included references.

- Add <refcontent> to allow better control of text in a reference.

- Add <link> to point to a resource related to the RFC.

1.2.2. New Attributes for Existing Elements

- Add "sortRefs", "symRefs", "tocDepth", and "tocInclude" attributes to <rfc> to cover processor instructions (PIs) that were in v2 that are still needed in the grammar.

- Add "ascii" attributes to <author>, <email>, <organization>, <street>, <city>, <region> and <code>. This allows an author to specify their information in their native scripts as the primary entry and still allow the ASCII-equivalent values to appear in the processed documents.

- Add "xml:lang" attribute to <artwork>, <postal>, and <author>. This is sometimes useful for renderers which display different fonts for ideographic characters used in China and Japan.

- Add the "section", "relative", and "sectionFormat" attributes to <xref>.

- Add the "numbered" and "removeinrfc" attributes to <section>.

- Add "autogeneratedBoilerplateText", "autogeneratedFigureNumber", "autogeneratedParagraphNumber", "autogeneratedSectionNumber", and "autogeneratedTableNumber" attributes to <rfc>, <figure>, <t>, <section>, and <texttable>, respectively.

1.2.3. Elements and Attributes Deprecated from v2

Deprecated elements and attributes are legacy vocabulary from v2 that are supported for input to v3 processors. They are likely to be removed from those processors in the future. Instead of being listed in Section 2, they are listed in in Section 3. See Appendix B for more information on processors and how they will handle deprecated features.

- Deprecate <list> in favor of <dl>, <ul>, and <ol>.

- Deprecate <spanx>; replace it with <strong>, <b>, <em>, <i>, and <tt>.

- Deprecate <vspace> because the major use for it, creating pseudo-paragraph-breaks in lists, is now handled properly.

- Deprecate <facsimile> because it is rarely used and is not actually useful; <email> is a much more useful way to get in touch with authors.

- Deprecate <format> because it is not useful and has caused surprise for authors in the past. If the goal is to provide a single URI for a reference, use the "target" attribute on <reference> instead.

- Deprecate the "title" attribute in <section>, <figure>, and <texttable> in favor of the new <titleelement>.

- Deprecate the "alt", and "src" attributes in <figure> because they overlap with the attributes in <artwork>.

- Deprecate the "xml:space" attribute in <artwork> because there was only one useful value. Deprecate "height" and "width" attribute in both <artwork> and <figure> because they are not needed for the new output formats.

- Deprecate the "pageno" attribute in <xref> because it was unused in v2. Deprecate the "none" and "title" values for the "format" attribute in <xref> because the former makes no sense semantically and the latter because it has unpredictable output.

1.2.4. Additional Changes from v2

- Allow non-ASCII characters in the format; the characters that are actually allowed will be determined by the RFC Editor.

- Allow <artwork> to be used on its own in <section> (no longer confine it to a figure).

- Give more specifics of handling the "type" attribute in <artwork>.

- In <address>, allow the sub-elements to be in any order.

- Allow <strong>, <b>, <em>, <i>, <tt>, <eref>, and <xref> in <cref>.

- Allow the sub-elements inside a <reference> to be in any order.

- Turned off the auto-generation of anchors in <cref> because there is no use case for them that cannot be achieved in other ways.

- Allow more than one <artwork>, or more than one <sourcecode>, in <figure>.

- In <front>, make <date> optional.

- In <postal>, allow the sub-elements to be in any order. Also allow the inclusion of the new <postalLine> instead of the older

elements.

- In <ttcol>, allow <xref>, <eref>, <iref>, and <cref> as optional children.

- Do not generate the grammar from a DTD, but instead get it directly from the Relax Next Generation (RNG) grammar [RNG].

1.3. Syntax Notation

The XML vocabulary here is defined in prose, based on the Relax NG schema ([RNC]) contained in Appendix E (specified in Relax NG Compact Notation, "RNC").

Note that the schema can be used for automated validity checks, but certain constraints are only described in prose (example: the conditionally required presence of the "abbrev" attribute).

2. Elements

The sections below describe all elements and their attributes.

Note that attributes not labeled "mandatory" are optional.

Many elements have an optional "anchor" attribute. In all cases, the value of the "anchor" attribute needs to be a valid XML "Name" (Section 2.3 of [XML]). In short, it is a text string with no spaces or colons; hyphens and underscores are allowed.

2.1. <abstract>

Contains the abstract of the document. See [RFCSTYLE] for more information on restrictions for the abstract.

This element appears as child element of: <front> (Section 2.26).

Content model:

One or more <t> elements (Section 2.53)

2.2. <address>

Provides address information for the author.

This element appears as child element of: <author> (Section 2.7).

Content model:

In any order:

- <postal> elements (Section 2.37)
- <phone> elements (Section 2.36)
- <facsimile> elements (Section 3.1)
- <email> elements (Section 2.23)
- <uri> elements (Section 2.60)

2.3. <annotation>

Provides additional prose augmenting a bibliographical reference.

For instance:

```
<annotation>
  Latest version available at <eref
  target='http://www.w3.org/TR/xml'/>.
</annotation>
```

...will generate the text used in the reference for [XML].

This element appears as child element of: <reference> (Section 2.42).

Content model:

In any order:

- Text
- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)
- <cref> elements (Section 2.16)
- <spanx> elements (Section 3.4)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)

- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.3.1. 'ascii' attribute

The ASCII equivalent of the annotation.

2.4. <area>

Provides information about the IETF area to which this document relates (currently not used when generating documents).

The value ought to be either the fullname or the abbreviation of one of the IETF areas as listed on <http://www.ietf.org/iesg/area.html>. The list at the time that this document is being published is: "Applications", "app", "General", "gen", "Internet", "int", "Operations and Management", "ops", "Real-time Applications and Infrastructure", "rai", "Routing", "rtg", "Security", "sec", "Transport", "tsv".

This element appears as child element of: <front> (Section 2.26).

Content model: only text content.

2.5. <artwork>

This element allows the inclusion of "artwork" into the document.

<artwork> provides full control of horizontal whitespace and line breaks, and thus is used for a variety of things, such as:

- diagrams ("line art"),
- complex tables, or
- protocol unit diagrams.

Alternatively, the "src" attribute allows referencing an external graphics file, such as a bitmap or a vector drawing, using a URI. In this case, the textual content acts as fallback for output representations that do not support graphics, and thus ought to

contain either a "line art" variant of the graphics, or otherwise prose that describes the included image in sufficient detail.

If the artwork includes either "&" or "<" characters, or the string "]]>" those characters need to be encoded using escaping or CDATA block(s); see <sourcecode> for a fuller description of these solutions.

In [XML2RFCv2], the <artwork> element was also used for source code and formal languages; in v3, this is now done with <sourcecode>.

There are at least four ways to include SVG in artwork in Internet Drafts:

- Inline, by including all of the SVG in the content of the element (such as "<artwork type="svg"><svg xmlns...">")

- Inline, but using XInclude (see Appendix B.1 (such as "<artwork type="svg"><xi:include href=...")

- As a data: URI (such as "<artwork type="svg" src="data:image/svg+xml,%3Csvg%20xmlns%3D%22http%3A%2F%2Fwww.w3...">")

- As a URI to an external entity (such as "<artwork type="svg" src="http://www.example.com/...">")

This element appears as child element of: <aside> (Section 2.6), <blockquote> (Section 2.11), <dd> (Section 2.18), <figure> (Section 2.25), <li> (Section 2.30), <section> (Section 2.46), and <t> (Section 2.53).

Content model:

Text

2.5.1. 'align' attribute

Controls whether the artwork appears left (default), centered, or right.

Allowed values:

- "left" (default)

- "center"

- "right"

2.5.2. 'alt' attribute

Alternative text description of the artwork (not just the caption). This is only used with artwork that is text-based; when the art comes from the "src" attribute, the alternative text comes from the text of the artwork itself.

2.5.3. 'height' attribute

Deprecated.

2.5.4. 'name' attribute

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

This attribute generally isn't used for document generation, but it can be helpful for other kinds of tools (such as automated syntax checkers which work by extracting the artwork).

2.5.5. 'src' attribute

The URI of a graphics file.

Note that this can be a "data" URI ([RFC2397]).

[[anchor7: Add an example of in-lined SVG here.]]

[[anchor8: Maybe include microlanguages for packets and call sequences if people want to create them.]]

2.5.6. 'type' attribute

Specifies the type of the artwork. The value of this attribute is free text with certain values designated as preferred. A private processor (as described in Appendix B) might add type-specific formatting to artwork with the preferred values. If a processor encounters a value for "type" that is not one of the preferred values, it can issue a warning but should still use the artwork as if it had no "type" attribute.

The preferred values for <artwork> types are:

- ascii-art

- call-flow

- hex-dump

- svg

The RFC Editor will maintain a complete list of the preferred values on its 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.

2.5.7. 'width' attribute

Deprecated.

2.5.8. 'xml:lang' attribute

Allows specification of the language used. This is sometimes useful for renderers which display different fonts for ideographic characters used in China and Japan.

2.5.9. 'xml:space' attribute

Deprecated.

2.6. <aside>

This element is a container for content that is semantically less important or tangential to the content that surrounds it. The elements inside this container are indented from both margins when displayed.

This element appears as child element of: <section> (Section 2.46).

Content model:

In any order:

- <t> elements (Section 2.53)

- <figure> elements (Section 2.25)

- <texttable> elements (Section 2.54)

- <iref> elements (Section 2.28)

- <artwork> elements (Section 2.5)

- <sourcecode> elements (Section 2.48)

2.7. <author>

Provides information about a document's author. This is used both for the document itself (at the beginning of the document) and for referenced documents (inside of <reference>).

The <author> elements contained within the document's <front> element are used to fill the boilerplate, and also to generate the "Author's Address" section (see [RFCSTYLE]).

Note that an "author" can also be just an organization (by not specifying any of the name attributes, but adding the <organization> child element).

Furthermore, the "role" attribute can be used to mark an author as "editor". This is reflected both on the front page and in bibliographical references. Note that this specification does not define a precise meaning for the term "editor".

See Section "Authors vs. Contributors" of [RFCPOLICY] for more information.

This element appears as child element of: <front> (Section 2.26).

Content model:

In this order:

1. One optional <organization> element (Section 2.35)

2. One optional <address> element (Section 2.2)

2.7.1. 'ascii' attribute

The ASCII equivalent of the author's full name.

2.7.2. 'fullname' attribute

The full name (used in the automatically generated "Author's Address" section).

2.7.3. 'initials' attribute

Author initials (used on the front page and in references).

The value contains one or more initials, each followed by a period. Initials should be provided as a whitespace separated list of pairs of a letter and a dot.

2.7.4. 'role' attribute

Specifies the role the author had in creating the document.

Allowed values:

- o "editor"

2.7.5. 'surname' attribute

The author's surname.

2.7.6. 'xml:lang' attribute

Allows specification of the language used. This is sometimes useful for renderers which display different fonts for CJK characters.

2.8. <b>

Causes the text to be displayed in bold. It is almost always a better idea to use the <strong> element instead. This element can be combined with other character formatting elements, and the formatting will be additive.

This element appears as child element of: <annotation> (Section 2.3), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), and <tt> (Section 2.57).

Content model:

In any order:

- o Text

- o <xref> elements (Section 2.62)

- o <eref> elements (Section 2.24)

- o <iref> elements (Section 2.28)

- o <cref> elements (Section 2.16)

- o <tt> elements (Section 2.57)

- <em> elements (Section 2.22)

- <i> elements (Section 2.27)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.9. <back>

Contains the "back" part of the document: the references and appendices. In <back>, <section> elements indicate appendices.

This element appears as child element of: <rfc> (Section 2.45).

Content model:

In this order:

1. Optional <displayreference> elements (Section 2.19)

2. Optional <references> elements (Section 2.43)

3. Optional <section> elements (Section 2.46)

2.10. <bcp14>

Marks text that are phrases defined in BCP 14 such as "MUST", "SHOULD NOT", and so on. When shown in some of the output representations, the text in this element might be highlighted. The use of this element is optional.

This element is only to be used around the actual phrase from BCP 14, not the full definition of a requirement. For example, it is correct to say "The packet <bcp14>MUST</bcp14> be dropped.", but it not correct to say "<bcp14>The packet MUST be dropped.</bcp14>".

Content model: only text content.

2.11. <blockquote>

Specifies a block of text is a quotation. The "cite" attribute is required, and must be a URI. [[anchor9: Should this element also use autogeneratedParagraphNumber? Or should there be an autogeneratedBlockquoteNumber?]]

This element appears as child element of: <section> (Section 2.46).

Content model:

In any order:

- Text
- <figure> elements (Section 2.25)
- <artwork> elements (Section 2.5)
- <sourcecode> elements (Section 2.48)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.11.1. 'anchor' attribute

Document-wide unique identifier for this quotation.

2.11.2. 'cite' attribute (mandatory)

The source of the citation. This must be a URI. [[anchor10: Needs an example of a cite for a reference that is already in the spec.]]

2.12. <c>

Provides the content of a cell in a table.

This element appears as child element of: <texttable> (Section 2.54).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- <spanx> elements (Section 3.4)

- <tt> elements (Section 2.57)

- <strong> elements (Section 2.50)

- <b> elements (Section 2.8)

- <em> elements (Section 2.22)

- <i> elements (Section 2.27)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.13. <city>

Gives the city name in a postal address.

This element appears as child element of: <postal> (Section 2.37).

Content model: only text content.

2.13.1. 'ascii' attribute

The ASCII equivalent of the city name.

2.14. <code>

Gives the postal region code.

This element appears as child element of: <postal> (Section 2.37).

Content model: only text content.

2.14.1. 'ascii' attribute

The ASCII equivalent of the postal code.

2.15. <country>

Gives the country in a postal address.

This element appears as child element of: <postal> (Section 2.37).

Content model: only text content.

2.16. <cref>

Represents a comment.

Comments can be used in a document while it is work-in-progress. They usually appear either inline and visually highlighted, at the end of the document (depending on file representation and settings of the processor), or not at all (when generating an RFC).

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <c> (Section 2.12), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <titleelement> (Section 2.56), <tt> (Section 2.57), and <ttcol> (Section 2.58).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <tt> elements (Section 2.57)

- <strong> elements (Section 2.50)

- <b> elements (Section 2.8)

- <em> elements (Section 2.22)

- <i> elements (Section 2.27)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.16.1. 'anchor' attribute

Document-wide unique identifier for this comment.

2.16.2. 'source' attribute

Holds the "source" of a comment, such as the name or the initials of the person who made the comment.

2.17. <date>

Provides information about the publication date.

Note that this element is used both for the boilerplate of the document being produced, and also inside bibliographic references that use the <front> element.

In the boilerplate case, it defines the date of publication for the current document (Internet Draft or RFC). When producing Internet-Drafts, the Draft Processor uses this date to compute the expiration date (see [IDGUIDE]). When one or more of "year", "month", or "day" are left out, the processor will attempt to use the current system date if the attributes that are present are consisten with that date.

Also in the first case, that month names, if given, need to match the full English month name: "January", "February", "March", "April", "May, "June", "July", "August", "September", "October", "November", or "December".

In the case of bibliographic references, 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.

This element appears as child element of: <front> (Section 2.26).

Content model: this element does not have any contents.

2.17.1. 'day' attribute

Day of publication; this is a number.

2.17.2. 'month' attribute

Month of publication; this is the English name of the month.

2.17.3. 'year' attribute

Year of publication.

2.18. <dd>

The definition part of an entry in a definition list.

This element appears as child element of: <dl> (Section 2.20).

Content model:

[[anchor11: Missing template for choice.]]

2.19. <displayreference>

This element gives a mapping between the anchor of a reference and a name that will be displayed instead. This allows authors to display more mneumonic anchor names for automatically-included references. For example, if the reference uses the anchor "RFC6949", the following would cause that anchor in the body of displayed documents to be "RFC-dev":

```
<displayreference from="RFC6449" to="RFC-dev"/>
```

This element appears as child element of: <back> (Section 2.9).

Content model: this element does not have any contents.

2.19.1. 'from' attribute (mandatory)

This attribute must be the name of an anchor in a <reference> element.

2.19.2. 'to' attribute (mandatory)

This attribute is a name that will be displayed as the anchor instead of the anchor that is given in the <reference> element. The string given must start with one of the following characters: 0-9, a-z, A-Z. The other characters in the string must be 0-9, a-z, A-Z, "-", ".", and "_".

2.20. <dl>

A definition list. Each entry has a pair of elements: a term (<dt>) and a definition (<dd>).

[[anchor12: The content model below is wrong in the -05 draft.]]

This element appears as child element of: <dd> (Section 2.18), <li> (Section 2.30), and <t> (Section 2.53).

Content model:

One or more sequences of:

1. One <dt> element

2. One <dd> element

2.20.1. 'hanging' attribute

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

Allowed values:

- "false" (default)

- "true"

2.20.2. 'spacing' attribute

Defines whether or not there is a blank line between entries. spacing="normal" indicates a single blank line, while spacing="compact" indicates no space between.

Allowed values:

- "normal" (default)

- "compact"

2.21. <dt>

The term being defined in a definition list.

This element appears as child element of: <dl> (Section 2.20).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)
- <cref> elements (Section 2.16)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.22. <em>

Indicates text that is semantically empahsized. This element will be displayed as italic after processing. This element has the same effects as <i>.

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), and <tt> (Section 2.57).

Content model:

In any order:

- Text
- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.23. <email>

Provides an email address.

The value is expected to be the scheme-specific part of a "mailto" URI (so does not include the prefix "mailto:"). See Section 2 of [RFC6068] for details.

This element appears as child element of: <address> (Section 2.2).

Content model: only text content.

2.23.1. 'ascii' attribute

The ASCII equivalent of the author's email address. This is only used if the email address has one or two internationalized components.

2.24. <eref>

Represents an "external" link (as specified in the "target" attribute). This can be used for creating linked references, such as in HTML output.

If the element has text content, that content will be used. Otherwise, the value of the target attribute will be inserted in angle brackets ([RFC3986], Appendix C).

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <titleelement> (Section 2.56), <tt> (Section 2.57), and <ttcol> (Section 2.58).

Content model: only text content.

2.24.1. 'target' attribute (mandatory)

URI of the link target (see Section 3 of [RFC3986]).

2.25. <figure>

This element is used to represent a figure, consisting of an optional preamble, the actual figure, an optional postamble, and an optional title.

This element appears as child element of: <aside> (Section 2.6), <blockquote> (Section 2.11), <dd> (Section 2.18), <li> (Section 2.30), <section> (Section 2.46), and <t> (Section 2.53).

Content model:

In this order:

1. One optional <titleelement> element (Section 2.56)

2. Optional <iref> elements (Section 2.28)

3. One optional <preamble> element (Section 2.40)

4. [[anchor13: Missing template for oneOrMore.]]

5. One optional <postamble> element (Section 2.39)

2.25.1. 'align' attribute

Used to change the alignment of <preamble> and <postamble>.

Note: does not affect title or <artwork> alignment.

Allowed values:

- "left" (default)

- "center"

- "right"

2.25.2. 'alt' attribute

Deprecated. If the goal is to provide a single URI for a reference, use the "target" attribute on <reference> instead.

2.25.3. 'anchor' attribute

Document-wide unique identifier for this figure.

Furthermore, the presence of this attribute causes the figure to be numbered.

2.25.4. 'autogeneratedFigureNumber' attribute

The number for this figure, if one is generated by the processor. This attribute and its value are automatically generated by the RFC Processor, and are ignored by other processors. If the value already exists when the RFC Processor is run, it is replaced.

2.25.5. 'height' attribute

Deprecated.

2.25.6. 'src' attribute

Deprecated.

2.25.7. 'suppress-title' attribute

Figures that have an "anchor" attribute will automatically get an autogenerated title (such as "Figure 1"), even if the "title" attribute and the <titleelement> element is absent. Setting this attribute to "true" will prevent this.

Allowed values:

- "true"

- "false" (default)

2.25.8. 'title' attribute

Deprecated. Use <titleelement> instead.

2.25.9. 'width' attribute

Deprecated.

2.26. <front>

Represent the "front matter": metadata (such as author information), abstract, and additional notes.

This element appears as child element of: <reference> (Section 2.42), and <rfc> (Section 2.45).

Content model:

In this order:

1. One <title> element (Section 2.55)

2. One or more <author> elements (Section 2.7)

3. One optional <date> element (Section 2.17)

4. Optional <area> elements (Section 2.4)

5. Optional <workgroup> elements (Section 2.61)

6. Optional <keyword> elements (Section 2.29)

7. One optional <abstract> element (Section 2.1)

8. Optional <note> elements (Section 2.33)

2.27. <i>

Causes the text to be displayed in italic. It is almost always a better idea to use the <em> element instead.

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), and <tt> (Section 2.57).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- <tt> elements (Section 2.57)

- <strong> elements (Section 2.50)

- <b> elements (Section 2.8)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.28. <iref>

Provides terms for the document's index.

Index entries can be either single items (when just the "item" attribute is given) or nested items (by specifying "subitem" as well).

For instance:

```
<iref item="Grammar" subitem="item"/>
```

will produce an index entry for "Grammar, item".

This element appears as child element of: <annotation> (Section 2.3), <aside> (Section 2.6), <b> (Section 2.8), <c> (Section 2.12), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <figure> (Section 2.25), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <section> (Section 2.46), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <titleelement> (Section 2.56), <tt> (Section 2.57), and <ttcol> (Section 2.58).

Content model: this element does not have any contents.

2.28.1. 'item' attribute (mandatory)

The item to include.

2.28.2. 'primary' attribute

Setting this to "true" declares the occurrence as "primary", which might cause it to be highlighted in the index.

Allowed values:

- "true"

- "false" (default)

2.28.3. 'subitem' attribute

The subitem to include.

2.29. <keyword>

Specifies a keyword applicable to the document.

Note that each element should only contain a single keyword; for multiple keywords, the element can simply be repeated.

Keywords are used both in the RFC Index and in the metadata of generated document representations.

This element appears as child element of: <front> (Section 2.26).

Content model: only text content.

2.30. <li>

A list element, used in <ol> and <ul>.

This element appears as child element of: <ol> (Section 2.34), and <ul> (Section 2.59).

Content model:

[[anchor14: Missing template for choice.]]

2.31. <link>

A link to an external document that is related to the RFC.

The following are the supported types of external documents that can be pointed to in a <link> element:

- The current ISSN for the RFC Series. The value for the "rel" attribute is "isPartOf". The link should use the form "urn:issn:".

- The DOI for this document. The value for the "rel" attribute is "describedBy". The link should use the form "doi:".

- The Internet Draft that was submitted to the RFC Editor to become the published RFC. The value for the "rel" attribute is "convertedFrom". The link should be to an IETF-controlled web site that retains copies of Internet Drafts.

- A representation of the document offered by the document author. The value for the "rel" attribute is "alternate". The link can be to a personally-run web site.

The RFC Processor needs to check the values for <link> before an RFC is published. The Draft Processor might remove some <link> elements during the draft submission process.

This element appears as child element of: <rfc> (Section 2.45).

Content model: this element does not have any contents.

2.31.1. 'href' attribute (mandatory)

The URI of the external document.

2.31.2. 'rel' attribute

The relationship of the external document to this one. The relationships are taken from Link Relations registry maintained by IANA [LINKRELATIONS].

2.32. <middle>

Represents the main content of the document.

This element appears as child element of: <rfc> (Section 2.45).

Content model:

One or more <section> elements (Section 2.46)

2.33. <note>

Creates an unnumbered section that appears after the abstract.

It is usually used for additional information to reviewers (working group information, mailing list, ...), or for additional publication information such as "IESG Notes".

This element appears as child element of: <front> (Section 2.26).

Content model:

One or more <t> elements (Section 2.53)

2.33.1. 'title' attribute (mandatory)

The title of the note.

2.34. <ol>

An ordered list. The labels on the items will be either a number or a letter, depending on the value of the style attribute.

This element appears as child element of: <dd> (Section 2.18), <li> (Section 2.30), and <t> (Section 2.53).

Content model:

One or more <li> elements (Section 2.30)

2.34.1. 'group' attribute

When a processor sees an <ol> element with a "group" attribute that has already been seen, the processor continues the numbering of the list from where the previous list with the same group name left off. If an <ol> element has both a "group" and "start" attribute, the group's numbering is reset to the given start value.

2.34.2. 'spacing' attribute

Defines whether or not there is a blank line between entries. spacing="normal" indicates a single blank line, while spacing="compact" indicates no space between.

Allowed values:

- "normal" (default)

- "compact"

2.34.3. 'start' attribute

The ordinal value to start the list at. This defaults to "1", and must be an integer of 0 or greater.

2.34.4. 'style' attribute

The style of the labels on list items. If the length of the style value is 1, the meaning is the same as it is for HTML:

a Lowercase letters (a, b, c, ...)

A Uppercase letters (A, B, C, ...)

1 Decimal numbers (1, 2, 3, ...)

i Lowercase Roman numerals (i, ii, iii, ...)

I Uppercase Roman numerals (I, II, III, ...)

[[anchor15: Need to determine, and then specify, what happens after the 26th letter.]]

If the length of the style value is greater than 1, the value must contain a percent-encoded indicator and other text. The value is a free-form text that allows counter values to be inserted using a "percent-letter" format. For instance, "[REQ%d]" generates labels of the form "[REQ1]", where "%d" inserts the item number as decimal number.

The following formats are supported:

%c Lowercase letters (a, b, c, ...)

%C Uppercase letters (A, B, C, ...)

%d Decimal numbers (1, 2, 3, ...)

%i Lowercase Roman numerals (i, ii, iii, ...)

%I Uppercase Roman numerals (I, II, III, ...)

%% Represents a percent sign

Other formats are reserved for future use.

It is an error for the style string to be empty, and processors will reject documents with an element that has such a style. If the

intention is to have a list that is indented without bullets or numbers, use <ul> with the 'empty="true"' attribute instead.

2.35. <organization>

Specifies the affiliation of an author.

This information appears in both the "Author's Address" section and on the front page (see [RFCSTYLE] for more information). If the value is long, an abbreviated variant can be specified in the "abbrev" attribute.

This element appears as child element of: <author> (Section 2.7).

Content model: only text content.

2.35.1. 'abbrev' attribute

Abbreviated variant.

2.35.2. 'ascii' attribute

The ASCII equivalent of the organization's name.

2.36. <phone>

Represents a phone number.

The value is expected to be the scheme-specific part of a "tel" URI (so does not include the prefix "tel:"), using the "global numbers" syntax. See Section 3 of [RFC3966] for details.

This element appears as child element of: <address> (Section 2.2).

Content model: only text content.

2.37. <postal>

Contains optional child elements providing postal information. These elements will be displayed in an order that is processor-specific. A postal address can contain only a set of <street>, <city>, <region>, <code>, and <country> elements, or only an ordered set of <postalLine> elements, but not both.

This element appears as child element of: <address> (Section 2.2).

Content model:

[[anchor16: Missing template for choice.]]

2.37.1. 'xml:lang' attribute

Allows specification of the language used. This is sometimes useful for renderers which display different fonts for CJK characters.

2.38. <postalLine>

Represents one line of a postal address. When more than one <postalLine> is given, the processor emits them in the order given.

This element appears as child element of: <postal> (Section 2.37).

Content model: only text content.

2.38.1. 'ascii' attribute

The ASCII equivalent of the text in the address line.

2.39. <postamble>

Gives text that appears at the bottom of a figure or table.

This element appears as child element of: <figure> (Section 2.25), and <texttable> (Section 2.54).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- <spanx> elements (Section 3.4)

- <tt> elements (Section 2.57)

- <strong> elements (Section 2.50)

- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.40. <preamble>

Gives text that appears at the top of a figure or table.

This element appears as child element of: <figure> (Section 2.25), and <texttable> (Section 2.54).

Content model:

In any order:

- Text
- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)
- <cref> elements (Section 2.16)
- <spanx> elements (Section 3.4)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.41. <refcontent>

Text that should appear between the title and the date of a reference. The purpose of this element is to prevent the need to abuse <seriesInfo> to get such text in a reference.

For example:

```
<reference anchor="April1">
  <front>
    <title>On Being A Fool</title>
    <author initials="K." surname="Phunny" fullname="Knot Phunny"/>
    <date year="2000" month="April"/>
  </front>
  <refcontent>Self-published pamphlet</refcontent>
</reference>
```

would render as:

```
   [April1]     Phunny, K., "On Being A Fool", Self-published
                pamphlet, April 2000.
```

This element appears as child element of: <reference> (Section 2.42).

Content model:

In any order:

- Text

- <tt> elements (Section 2.57)

- <b> elements (Section 2.8)

- <i> elements (Section 2.27)

- <em> elements (Section 2.22)

- <strong> elements (Section 2.50)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.41.1. 'ascii' attribute

The ASCII equivalent of the added text.

2.42. <reference>

Represents a bibliographical reference.

This element appears as child element of: <references> (Section 2.43).

Content model:

In this order:

1. One <front> element (Section 2.26)

2. In any order:

   * <seriesInfo> elements (Section 2.47)

   * <format> elements (Section 3.2)

   * <refcontent> elements (Section 2.41)

   * <annotation> elements (Section 2.3)

2.42.1. 'anchor' attribute (mandatory)

Document-wide unique identifier for this reference. Usually, this will be used both to "label" the reference in the references section, and as an identifier in links to this reference entry.

2.42.2. 'target' attribute

Holds the URI for the reference.

Note that depending on the <seriesInfo> element, a URI might not be needed, nor desirable, as it can be automatically generated (for instance, for RFCs).

2.43. <references>

Contains a set of bibliographical references.

In the early days of the RFC series, there was only one "References" section per RFC. This convention was later changed to group references into two sets, "Normative" and "Informative" as described

in [RFCSTYLE]). This vocabulary supports the split with the "title" attribute.

This element appears as child element of: <back> (Section 2.9).

Content model:

One or more <reference> elements (Section 2.42)

2.43.1. 'title' attribute

Provides the title for the References section (defaulting to "References").

In general, the title should be either "Normative References" or "Informative References".

2.44. <region>

Provides the region name in a postal address.

This element appears as child element of: <postal> (Section 2.37).

Content model: only text content.

2.44.1. 'ascii' attribute

The ASCII equivalent of the region name.

2.45. <rfc>

This is the root element of the xml2rfc vocabulary.

Processors distinguish between RFC mode ("number" attribute being present) and Internet-Draft mode ("docName" attribute being present): it is invalid to specify both. Setting neither "number" nor "docName" can be useful for producing other types of document but is out-of-scope for this specification.

Content model:

In this order:

1. One optional <link> element (Section 2.31)

2. One <front> element (Section 2.26)

3. One <middle> element (Section 2.32)

4. One optional <back> element (Section 2.9)

2.45.1. 'autogeneratedBoilerplateText' attribute

The full boilerplate text for this document. This attribute and its value are automatically generated by the RFC Processor, and are ignored by other processors. If the value already exists when the RFC Processor is run, it is replaced.

2.45.2. 'category' attribute

Document category (see Appendix A.1).

Allowed values:

- "std"

- "bcp"

- "info"

- "exp"

- "historic"

2.45.3. 'consensus' attribute

Affects the generated boilerplate.

See [RFC5741] for more information.

Allowed values:

- "no"

- "yes"

2.45.4. 'docName' attribute

Specifies the name or the Internet Draft or private document. A private document is one whose representation is not meant to be an Internet Draft or RFC. If the value for this attribute starts with the string "draft-", a processor treats the input as an Internet Draft; if the value starts with any other string, a processor treats this as a private document. The name in the value should be the document name without any file extension. That is, the value for

this attribute should be "draft-ietf-somewg-someprotocol-07", not "draft-ietf-somewg-someprotocol-07.txt".

A processor should give an error if both the "docName" and "number" attributes are given in the <rfc> element.

2.45.5. 'ipr' attribute

Represents the Intellectual Property status of the document. See Appendix A.2 for details.

Allowed values:

- "full2026"
- "noDerivativeWorks2026"
- "none"
- "full3667"
- "noModification3667"
- "noDerivatives3667"
- "full3978"
- "noModification3978"
- "noDerivatives3978"
- "trust200811"
- "noModificationTrust200811"
- "noDerivativesTrust200811"
- "trust200902"
- "noModificationTrust200902"
- "noDerivativesTrust200902"
- "pre5378Trust200902"

2.45.6. 'iprExtract' attribute

Identifies a single section within the document for which extraction "as-is" is explicitly allowed (only relevant for historic values of the "ipr" attribute).

2.45.7. 'number' attribute

The number of the RFC to be produced by a processor.

A processor should give an error if both the "docName" and "number" attributes are given in the <rfc> element.

2.45.8. 'obsoletes' attribute

A comma-separated list of RFC numbers or Internet-Draft names.

Processors ought to parse the attribute value, so that incorrect references can be detected and, depending on output representation, hyperlinks can be generated. Also, the value ought to be reformatted to insert whitespace after each comma if not already present.

2.45.9. 'seriesNo' attribute

When producing a document within document series (such as "STD"): the number within that series.

2.45.10. 'sortRefs' attribute

Specifies whether or not a processor will sort the references in each reference section.

Allowed values:

o "yes"

o "no" (default)

2.45.11. 'submissionType' attribute

The document stream.

See Section 2 of [RFC5741] for details.

Allowed values:

o "IETF" (default)

- "IAB"

- "IRTF"

- "independent"

2.45.12. 'symRefs' attribute

Specifies whether or not a processor will use symbolic references (such as "[RFC2119]"). If the value for this is "no", the references come out as numbers (such as "[3]").

Allowed values:

- "yes" (default)

- "no"

2.45.13. 'tocDepth' attribute

Specifies number of levels of heading for a processor to include in the table of contents; the default is "3".

2.45.14. 'tocInclude' attribute

Specifies whether or not a processor will include a table of contents in generated files.

Allowed values:

- "yes" (default)

- "no"

2.45.15. 'updates' attribute

A comma-separated list of RFC numbers or Internet-Draft names.

Processors ought to parse the attribute value, so that incorrect references can be detected and, depending on output representation, hyperlinks can be generated. Also, the value ought to be reformatted to insert whitespace after each comma if not already present.

2.45.16. 'xml:lang' attribute

The natural language used in the document (defaults to "en").

See Section 2.12 of [XML] for more information.

2.46. <section>

Represents a section (when inside a <middle> element) or an appendix (when inside a <back> element).

Sub-sections are created by nesting <section> elements inside <section> elements. Sections are allowed to be empty.

This element appears as child element of: <back> (Section 2.9), <middle> (Section 2.32), and <section> (Section 2.46).

Content model:

In this order:

1. One optional <titleelement> element (Section 2.56)

2. In any order:

   * <t> elements (Section 2.53)

   * <aside> elements (Section 2.6)

   * <blockquote> elements (Section 2.11)

   * <figure> elements (Section 2.25)

   * <texttable> elements (Section 2.54)

   * <iref> elements (Section 2.28)

   * <artwork> elements (Section 2.5)

   * <sourcecode> elements (Section 2.48)

3. Optional <section> elements (Section 2.46)

2.46.1. 'anchor' attribute

Document-wide unique identifier for this section.

2.46.2. 'autogeneratedSectionNumber' attribute

The number for this section, if one is generated by the processor. This attribute and its value are automatically generated by the RFC Processor, and are ignored by other processors. If the value already exists when the RFC Processor is run, it is replaced.

2.46.3. 'numbered' attribute

If set to "no", this section does not get a section number. Processors will verify that such a section is not followed by a numbered section in a part, and will verify that the section is a top-level section.

Allowed values:

- "yes" (default)

- "no"

2.46.4. 'removeInRFC' attribute

If set to "yes", this section is marked in the processor with text indicating that it should be removed before the document is published as an RFC.

Allowed values:

- "yes"

- "no" (default)

2.46.5. 'title' attribute

Deprecated. Use <titleelement> instead.

2.46.6. 'toc' attribute

Indicates to a processor whether or not the section is to be included in the table of contents. This only takes effect if the level of the section would have appeared in the table of contents based on the "tocDepth" attribute of the <rfc> element, and of course only if the table of contents is being created based on the "tocInclude" attribute of the <rfc> element. If this is set to "exclude", any section below this one will be excluded as well. The "default" value indicates to include the section if it would be included by the tocDepth attribute of the <rfc> element.

Allowed values:

- "include"

- "exclude"

- "default" (default)

2.47. <seriesInfo>

Specifies the document series in which this document appears, and also specifies an identifier within that series.

This element appears as child element of: <reference> (Section 2.42).

Content model: this element does not have any contents.

2.47.1. 'asciiname' attribute

The ASCII equivalent of the name field.

2.47.2. 'asciivalue' attribute

The ASCII equivalent of the value field.

2.47.3. 'name' attribute (mandatory)

The name of the series.

The following names trigger specific processing (such as for auto-generating links, and adding descriptions such as "work in progress"): "BCP", "FYI", "Internet-Draft", "RFC", and "STD".

2.47.4. 'value' attribute (mandatory)

The identifier within the series specified by the "name" attribute.

For BCPs, FYIs, RFCs, and STDs this is the number within the series. For Internet-Drafts, it is the full draft name (ending with the two-digit version number).

2.48. <sourcecode>

This element allows the inclusion of sourcecode into the document.

<sourcecode> provides full control of horizontal whitespace and line breaks. It is thus useful for source code and formal languages (such as ABNF or the RNC notation used in this document).

For artwork such as character-based art, diagrams of message layouts, and so on, use the <artwork> element instead.

A common problem authors have with <sourcecode> is that the XML processor returns errors if the text in the artwork contains either

the "&" or "<" character, or the string "]]>". To avoid these problems, the "&" and "<" characters may be escaped using the strings "&amp;" and "&lt;", respectively; the "]]>" string can be represented as "]]&gt;". Alternatively, they may be surrounded in a CDATA structure: "<![CDATA[]]>". For example:

```
Desired output:
   allowed-chars = "." | "," | "&" | "<" | ">" | "|"

Using escaping:
<sourcecode>
   allowed-chars = "." | "," | "&amp;" | "&lt;" | "&gt;" | "|"
</sourcecode>

Using CDATA:
<sourcecode>
<![CDATA[   allowed-chars = "." | "," | "&" | "<" | ">" | "|"]]>
</sourcecode>
```

Using CDATA is not a panacea, but it does help prevent having to use escapes in places where using using escapes can cause other problems, such as difficulty of inclusion from other documents.

This element appears as child element of: <aside> (Section 2.6), <blockquote> (Section 2.11), <dd> (Section 2.18), <figure> (Section 2.25), <li> (Section 2.30), <section> (Section 2.46), and <t> (Section 2.53).

Content model: only text content.

2.48.1. 'name' attribute

A filename suitable for the contents (such as for extraction to a local file). This attribute generally isn't used for document generation, but it can be helpful for other kinds of tools (such as automated syntax checkers which work by extracting the source code).

2.48.2. 'type' attribute

Specifies the type of the sourcecode. The value of this attribute is free text with certain values designated as preferred. A private processor (as described in Appendix B) might add type-specific formatting to sourcecode with the preferred values. If a processor encounters a value for "type" that is not one of the preferred values, it can issue a warning but should still use the artwork as if it had no "type" attribute.

The preferred values for <sourcecode> types are:

- abnf
- asn.1
- bash
- c++
- c
- cbor
- dtd
- java
- javascript
- json
- mib
- perl
- pseudocode
- python
- rnc
- xml

The RFC Editor will maintain a complete list of the preferred values on its 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.

2.49. <street>

Provides a street address.

This element appears as child element of: <postal> (Section 2.37).

Content model: only text content.

2.49.1. 'ascii' attribute

The ASCII equivalent of the street address.

2.50. <strong>

Indicates text that is semantically strong. This element will be displayed as bold after processing. This element has the same effects as <b>. This element can be combined with other character formatting elements, and the formatting will be additive.

This element appears as child element of: <annotation> (Section 2.3), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), and <tt> (Section 2.57).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- <tt> elements (Section 2.57)

- <em> elements (Section 2.22)

- <i> elements (Section 2.27)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.51. <sub>

Causes the text to be displayed as subscript, approximately half a letter-height lower than normal text. This element can be combined with other character formatting elements, and the formatting will be

additive.

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <strong> (Section 2.50), <t> (Section 2.53), and <tt> (Section 2.57).

Content model:

In any order:

- Text
- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)
- <cref> elements (Section 2.16)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)

2.52. <sup>

Causes the text to be displayed as superscript, approximately half a letter-height higher than normal text. This element can be combined with other character formatting elements, and the formatting will be additive.

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <strong> (Section 2.50), <t> (Section 2.53), and <tt> (Section 2.57).

Content model:

In any order:

- Text
- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)
- <cref> elements (Section 2.16)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)

2.53. <t>

Contains a paragraph of text.

This element appears as child element of: <abstract> (Section 2.1), <aside> (Section 2.6), <dd> (Section 2.18), <li> (Section 2.30), <list> (Section 3.3), <note> (Section 2.33), and <section> (Section 2.46).

Content model:

In any order:

- Text
- <list> elements (Section 3.3)
- <ol> elements (Section 2.34)
- <ul> elements (Section 2.59)
- <dl> elements (Section 2.20)

- <figure> elements (Section 2.25)
- <artwork> elements (Section 2.5)
- <sourcecode> elements (Section 2.48)
- <xref> elements (Section 2.62)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.28)
- <cref> elements (Section 2.16)
- <spanx> elements (Section 3.4)
- <vspace> elements (Section 3.5)
- <tt> elements (Section 2.57)
- <strong> elements (Section 2.50)
- <b> elements (Section 2.8)
- <em> elements (Section 2.22)
- <i> elements (Section 2.27)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)

2.53.1. 'anchor' attribute

Document-wide unique identifier for this paragraph.

2.53.2. 'autogeneratedParagraphNumber' attribute

The number for this paragraph, if one is generated by the processor. This attribute and its value are automatically generated by the RFC Processor, and are ignored by other processors. If the value already exists when the RFC Processor is run, it is replaced.

2.53.3. 'hangText' attribute

Deprecated. Instead use <dd> inside of a definition list (<dl>).

2.54. <texttable>

Contains a table, consisting of an optional preamble, a header line, rows, an optional postamble, and an optional title.

The number of columns in the table is determined by the number of <ttcol> elements. The number of rows in the table is determined by the number of <c> elements divided by the number of columns. There is no requirement that the number of <c> elements be evenly divisible by the number of columns.

This element appears as child element of: <aside> (Section 2.6), and <section> (Section 2.46).

Content model:

In this order:

1. One optional <titleelement> element (Section 2.56)

2. One optional <preamble> element (Section 2.40)

3. One or more <ttcol> elements (Section 2.58)

4. Optional <c> elements (Section 2.12)

5. One optional <postamble> element (Section 2.39)

2.54.1. 'align' attribute

Determines the horizontal alignment of the table.

Allowed values:

- "left"

- "center" (default)

- "right"

2.54.2. 'anchor' attribute

Document-wide unique identifier for this table.

Furthermore, the presence of this attribute causes the table to be numbered.

2.54.3. 'autogeneratedTableNumber' attribute

The number for this table, if one is generated by the processor. This attribute and its value are automatically generated by the RFC Processor, and are ignored by other processors. If the value already exists when the RFC Processor is run, it is replaced.

2.54.4. 'style' attribute

Selects which borders should be drawn, where

- "all" means borders around all table cells,
- "full" is like "all" except no horizontal lines between table rows (except below the column titles),
- "headers" adds just a separator between column titles and rows, and
- "none" means no borders at all.

Allowed values:

- "all"
- "none"
- "headers"
- "full" (default)

2.54.5. 'suppress-title' attribute

Tables that have an "anchor" attribute will automatically get an autogenerated title (such as "Table 1"), even if the "title" attribute is absent. Setting this attribute to "true" will prevent this.

Allowed values:

- "true"
- "false" (default)

2.54.6. 'title' attribute

Deprecated. Use <titleelement> instead.

2.55. <title>

Represents the document title.

When this element appears in the <front> element of the current document, the title might also appear in page headers or footers. If it is long (~40 characters), the "abbrev" attribute is used to specify an abbreviated variant.

This element appears as child element of: <front> (Section 2.26).

Content model: only text content.

2.55.1. 'abbrev' attribute

Specifies an abbreviated variant of the document title.

2.55.2. 'ascii' attribute

The ASCII equivalent of the title.

2.56. <titleelement>

The title of the section, figure, or texttable. This title can have flow markup such as to make some characters use a fixed-width font, or to include references.

This element appears as child element of: <figure> (Section 2.25), <section> (Section 2.46), and <texttable> (Section 2.54).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- <tt> elements (Section 2.57)

2.57. <tt>

Causes the text to be displayed in a constant-width font. This element can be combined with other character formatting elements, and the formatting will be additive.

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <blockquote> (Section 2.11), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <refcontent> (Section 2.41), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), and <titleelement> (Section 2.56).

Content model:

In any order:

- Text

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- <strong> elements (Section 2.50)

- <b> elements (Section 2.8)

- <em> elements (Section 2.22)

- <i> elements (Section 2.27)

- <sub> elements (Section 2.51)

- <sup> elements (Section 2.52)

2.58. <ttcol>

Contains a column heading in a table.

This element appears as child element of: <texttable> (Section 2.54).

Content model:

In any order:

- <xref> elements (Section 2.62)

- <eref> elements (Section 2.24)

- <iref> elements (Section 2.28)

- <cref> elements (Section 2.16)

- Text

2.58.1. 'align' attribute

Determines the horizontal alignment within the table column.

Allowed values:

- "left" (default)

- "center"

- "right"

2.58.2. 'width' attribute

The desired column width (as integer 0..100 followed by "%").

2.59. <ul>

An unordered list. The labels on the items will be symbols picked by the processor.

This element appears as child element of: <dd> (Section 2.18), <li> (Section 2.30), and <t> (Section 2.53).

Content model:

One or more <li> elements (Section 2.30)

2.59.1. 'empty' attribute

Defines whether or not the label is empty. empty="true" indicates that no label be shown.

Allowed values:

- o "false" (default)

- o "true"

2.59.2. 'spacing' attribute

Defines whether or not there is a blank line between entries. spacing="normal" indicates a single blank line, while spacing="compact" indicates no space between.

Allowed values:

- o "normal" (default)

- o "compact"

2.60. <uri>

Contains a web address associated with the author.

The contents should be a valid URI (see Section 3 of [RFC3986]).

This element appears as child element of: <address> (Section 2.2).

Content model: only text content.

2.61. <workgroup>

This element is used to specify the Working Group (IETF) or Research Group (IRTF) from which the document originates, if any. The recommended format is the official name of the Working Group (with some capitalization).

In Internet-Drafts, this is used in the upper left corner of the boilerplate, replacing the "Network Working Group" string. Formatting software can append the words "Working Group" or "Research Group", depending on the "submissionType" property on the <rfc> element (Section 2.45.11).

This element appears as child element of: <front> (Section 2.26).

Content model: only text content.

2.62. <xref>

Inserts a reference to a different part of a document.

The generated text depends on whether the <xref> is empty (in which

case the processor will try to generate a meaningful text fragment), and the nature of the referenced document part. [[anchor17: Need to say more about the processor output if the element is empty and if it has text.]]

Any element that allows the "anchor" attribute can be referenced, however there are restrictions with respect to the text content being generated. For instance, a <t> can be a reference target, however, because paragraphs are not (visibly) numbered, the author will have to make sure that the prose is sufficient for a reader to understand what is being referred to.

This element appears as child element of: <annotation> (Section 2.3), <b> (Section 2.8), <c> (Section 2.12), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <i> (Section 2.27), <li> (Section 2.30), <postamble> (Section 2.39), <preamble> (Section 2.40), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <titleelement> (Section 2.56), <tt> (Section 2.57), and <ttcol> (Section 2.58).

Content model: only text content.

2.62.1. 'format' attribute

This attribute is used to control the format of the generated reference text.

"counter"

Inserts just a counter; this is used for the number of a section, figure, or table. For example:

```
<section anchor="overview">Protocol Overview</section>
. . .
See Section <xref target="overview" format="counter"/>
for an overview.
```

might generate

```
See Section 1.7 for an overview.
```

"default"

Inserts a text fragment that describes the referenced part completely, such as "Section 2" or "Table 4" for internal links, or "[XML]" for links to references. For example:

```
<section anchor="overview">Protocol Overview</section>
. . .
See <xref target="overview"/> for an overview.
```

might generate

```
See Section 1.7 for an overview.
```

"none"

There will be no auto-generated text. This is useful for creating a link that will appear in the processed text.

"title"

Inserts a title for the referenced element (usually obtained from the referenced element's "title" attribute; some processors also use the <title> child element or a <reference> target).

Allowed values:

- "counter"
- "title"
- "none"
- "default" (default)

2.62.2. 'pageno' attribute

Deprecated.

Allowed values:

- "true"
- "false" (default)

2.62.3. 'relative' attribute

Specifies a relative reference from the main target. [[anchor18: Need more description here about how this will be displayed.]]

2.62.4. 'section' attribute

Specifies a section for the generated reference. For example,

```
     See <xref section="2.3" target="RFC6949"/> for more information.
```

would generate

```
     See Section 2.3 of [RFC6949] for more information.
```

2.62.5. 'sectionFormat' attribute

The format that the section reference will be displayed in. The acceptable values are:

- "of" (default)

- "comma"

- "parens"

For example:

```
  See <xref target="RFC6949" section="2.3" sectionFormat="of"/>
  for more information.
  See <xref target="RFC6949" section="2.4" sectionFormat="comma"/>
  for more information.
  See <xref target="RFC6949" section="2.5" sectionFormat="parens"/>
  for more information.
```

would generate

```
     See Section 2.3 of [RFC6949] for more information.
     See [RFC6949], Section 2.4 for more information.
     See [RFC6949] (Section 2.5) for more information.
```

2.62.6. 'target' attribute (mandatory)

Identifies the document component being referenced.

The value needs to match the value of the "anchor" attribute of another element in the document.

3. Elements from v2 That Have Been Deprecated

This section lists the elements from v2 that have been deprecated. Note that some elements in v3 have attributes from v2 that are deprecated; those are not listed here.

3.1. <facsimile>

Deprecated. The <email> element is a much more useful way to get in touch with authors.

This element appears as child element of: <address> (Section 2.2).

Content model: only text content.

3.2. <format>

Deprecated. If the goal is to provide a single URI for a reference, use the "target" attribute on <reference> instead.

This element appears as child element of: <reference> (Section 2.42).

Content model: this element does not have any contents.

3.2.1. 'octets' attribute

Deprecated.

3.2.2. 'target' attribute

Deprecated.

3.2.3. 'type' attribute (mandatory)

Deprecated.

3.3. <list>

Deprecated. Instead, use <dl> for list/@style "hanging"; <ul> for list/@style "empty" or "symbols"; and <ol> for list/@style "letters", "numbers", "counter", or "format".

This element appears as child element of: <t> (Section 2.53).

Content model:

One or more <t> elements (Section 2.53)

3.3.1. 'counter' attribute

Deprecated. The functionality of this attribute has been replaced with <ol>/@start.

3.3.2. 'hangIndent' attribute

Deprecated. Use <dl> instead.

3.3.3. 'style' attribute

Deprecated.

3.4. <spanx>

Deprecated.

This element appears as child element of: <annotation> (Section 2.3), <c> (Section 2.12), <postamble> (Section 2.39), <preamble> (Section 2.40), and <t> (Section 2.53).

Content model: only text content.

3.4.1. 'style' attribute

Deprecated. Instead of <spanx style="emph">, use <b> instead of <spanx style="strong">, use <i>; instead of <spanx style="verg">, use <tt>.

3.4.2. 'xml:space' attribute

Deprecated.

Allowed values:

- "default"

- "preserve" (default)

3.5. <vspace>

Deprecated. In earlier versions of this format, <vspace> was often used to get an extra blank line in a list element; in the v3 vocabulary, that can be done instead by using multiple <t> elements inside the <li> element. Other uses have no direct replacement.

This element appears as child element of: <t> (Section 2.53).

Content model: this element does not have any contents.

3.5.1. 'blankLines' attribute

Deprecated.

4. Internationalization Considerations

This format is based on [XML], thus does not have any issues representing arbitrary Unicode [UNICODE] characters in text content. The RFC Editor may restrict some of the characters that can be used in a particular RFC; the rules for such restrictions are covered in [[anchor19: some other document to be named later when it is stable]].

5. Security Considerations

The "name" attribute on the <artwork> element (Section 2.5.4) can be used to derive a filename for saving to a local file system. Trusting this kind of information without pre-processing is a known security risk; see Section 4.3 of [RFC6266] for more information.

The "type" attribute of the <artwork> and <sourcecode> elements is meant to encourage processors to automatically extract known types of content from an RFC or Internet Draft. While extraction is probably safe, those processors might also think that they could further process the extracted content such as by rendering artwork or executing code. Doing so without first sanity-checking the extracted content is clearly a terrible idea from a security perspective. More generally, a processor that is reading RFCs or Internet Drafts needs to be suspicious of any content that it intends to post-process.

All security considerations related to XML processing are relevant as well (see Section 7 of [RFC3470]).

6. IANA Considerations

6.1. Internet Media Type Registration

IANA maintains the registry of Internet media types [BCP13] at <http://www.iana.org/assignments/media-types>.

This document updates the specification for the Internet media type "application/rfc+xml" from the one in [XML2RFCv2]. The following is to be registered with IANA.

Type name: application

Subtype name: rfc+xml

Required parameters: There are no required parameters.

Optional parameters: "charset": This parameter has identical semantics as the charset parameter of the "application/xml" media type specified in [RFC3023].

Encoding considerations: Identical to those of "application/xml" as described in Section 3.2 of [RFC3023].

Security considerations: As defined in Section 5. In addition, as this media type uses the "+xml" convention, it inherits the security considerations described in Section 10 of [RFC3023].

Interoperability considerations: N/A

Published specification: This specification.

Applications that use this media type: Applications that transform xml2rfc to output representations such as plain text or HTML, plus additional analysis tools.

Fragment identifier considerations: The "anchor" attribute is used for assigning document-wide unique identifiers that can be used as shorthand pointers, as described in Section 2.8 of [XPOINTER].

Additional information:

Deprecated alias names for this type: None.

Magic number(s): As specified for "application/xml" in Section 3.2 of [RFC3023].

File extension(s): .xml or .rfcxml when disambiguation from other XML files is needed

Macintosh file type code(s): TEXT

Person & email address to contact for further information: See Authors Section.

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Authors Section.

Change controller: RFC Series Editor (rse@rfc-editor.org)

6.2. Link Relation Registration

The following is a proposed addition to [LINKRELATIONS].

Relation Name: convertedFrom

Description: The document linked to was later converted to the document that contains this link relation. For example, an RFC can have a link to the Internet Draft that became the RFC; in that case, the link relation would be "convertedFrom".

Reference: This document.

Notes: This relation is different than "predecessor-version" in that "predecessor-version" is for items in a version control system. It is also different that "previous" in that this relation is used for converted resources, not those that are part of a sequence of resources.

Application Data: none

7. Acknowledgments

Thanks to everybody who reviewed this document and provided feedback and/or specification text. Thanks especially go to Julian Reschke for editing [XML2RFCv2] and those who provided feedback on that document.

We also thank Marshall T. Rose for both the original design and the reference implementation of the "xml2rfc" processor.

8. References

8.1. Normative References

[XML] Maler, E., Yergeau, F., Paoli, J., Sperberg-McQueen, M., and T. Bray, "Extensible Markup Language (XML) 1.0 (Fifth Edition)", W3C Recommendation REC-xml-20081126, November 2008, <http://www.w3.org/TR/2008/REC-xml-20081126/>.

Latest version available at

<http://www.w3.org/TR/xml>.

[XML2RFCv2] Reschke, J., "The 'XML2RFC' version 2 Vocabulary", draft-reschke-xml2rfc (work in progress), 2014.

8.2. Informative References

[BCP13] Freed, N., Klensin, J., and T. Hansen, "Media Type Specifications and Registration Procedures", BCP 13, RFC 6838, January 2013.

[IDGUIDE] Housley, R., "Guidelines to Authors of Internet-Drafts", <http://www.ietf.org/id-info/guidelines.html>.

[LINKRELATIONS] IANA, "Link Relations", <http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

[RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", BCP 9, RFC 2026, October 1996.

[RFC2397] Masinter, L., "The "data" URL scheme", RFC 2397, August 1998.

[RFC2629] Rose, M., "Writing I-Ds and RFCs using XML", RFC 2629, June 1999.

[RFC3023] Murata, M., St. Laurent, S., and D. Kohn, "XML Media Types", RFC 3023, January 2001.

[RFC3470] Hollenbeck, S., Rose, M., and L. Masinter, "Guidelines for the Use of Extensible Markup Language (XML) within IETF Protocols", BCP 70, RFC 3470, January 2003.

[RFC3966] Schulzrinne, H., "The tel URI for Telephone Numbers", RFC 3966, December 2004.

[RFC3986] Berners-Lee, T., Fielding, R., and L. Masinter, "Uniform Resource Identifier (URI): Generic Syntax", STD 66, RFC 3986, January 2005.

[RFC5741] Daigle, L. and O. Kolkman, "RFC Streams, Headers, and Boilerplates", RFC 5741, December 2009.

[RFC6068] Duerst, M., Masinter, L., and J. Zawinski, "The 'mailto' URI Scheme", RFC 6068, October 2010.

[RFC6266] Reschke, J., "Use of the Content-Disposition Header Field in the Hypertext Transfer Protocol (HTTP)", RFC 6266, June 2011.

[RFC6949] Flanagan, H. and N. Brownlee, "RFC Series Format Requirements and Future Development", RFC 6949, May 2013.

[RFCPOLICY] RFC Editor, "RFC Editorial Guidelines and Procedures", <http://www.rfc-editor.org/policy.html>.

[RFCSTYLE] Heather, H. and S. Ginoza, "RFC Style Guide", draft-iab-styleguide (work in progress).

[RNC] Clark, J., "RELAX NG Compact Syntax", OASIS , November 2002, <http://www.oasis-open.org/committees/relax-ng/compact-20021121.html>.

[RNG] ISO/IEC, "Information Technology - Document Schema Definition Languages (DSDL) - Part 2: Regular-Grammar-Based Validation - RELAX NG. Second Edition.", ISO/IEC 19757-2:2008(E), December 2008.

A useful source of RNG-related information is <http://relaxng.org/>.

[TLP2.0] IETF Trust, "Legal Provisions Relating to IETF Documents", February 2009, <http://trustee.ietf.org/license-info/IETF-TLP-2.htm>.

[TLP3.0] IETF Trust, "Legal Provisions Relating to IETF Documents", September 2009, <http://trustee.ietf.org/license-info/IETF-TLP-3.htm>.

[TLP4.0] IETF Trust, "Legal Provisions Relating to IETF Documents", December 2009, <http://trustee.ietf.org/license-info/IETF-TLP-4.htm>.

[UNICODE] The Unicode Consortium, "The Unicode Standard, Version 6.3.0", September 2013, <http://www.unicode.org/versions/Unicode6.3.0/>.

[XInclude] Marsh, J., Orchard, D., and D. Veillard, "XML Inclusions (XInclude) Version 1.0 (Second Edition)", W3C Recommendation REC-xinclude-20061115, November 2006, <http://www.w3.org/TR/xinclude/>.

Latest version available at
<http://www.w3.org/TR/xinclude/>.

[XPOINTER] Grosso, P., Maler, E., Marsh, J., and N. Walsh, "XPointer Framework", W3C Recommendation REC-xptr-framework-20030325, March 2003, <http://www.w3.org/TR/2003/REC-xptr-framework-20030325/>.

Latest version available at
<http://www.w3.org/TR/xptr-framework/>.

Appendix A. Front Page Generation

A.1. The /rfc/@category Attribute

For RFCs, the category determines the "maturity level" (see Section 4 of [RFC2026]). The allowed values are "std" for "Standards Track", "bcp" for "BCP", "info" for "Informational", "exp" for "Experimental", and "historic" for "Historic".

For Internet-Drafts, the category attribute is not needed, but will appear on the front page as "Intended Status". Supplying this information can be useful to reviewers.

A.2. The /rfc/@ipr Attribute

This attribute value can take a long list of values, each of which describes an IPR policy for the document. This attribute's values are not the result of a grand plan, but remain simply for historic reasons. Of these values, only a few are currently in use; all others are supported by the various tools for backwards compatibility with old source files.

> Note: some variations of the boilerplate are selected based on the document's date; therefore it is important to specify the "year", "month" and "day" attributes of the <date> element when archiving the XML source of an Internet-Draft on the day of submission.

Disclaimer: THIS ONLY PROVIDES IMPLEMENTATION INFORMATION. IF YOU NEED LEGAL ADVICE, PLEASE CONTACT A LAWYER. For further information, refer to <http://trustee.ietf.org/docs/IETF-Copyright-FAQ.pdf>.

For the current "Status Of This Memo" text, the submissionType attribute determines whether a statement about "Code Components" is inserted (which is the case for the value "IETF", which is the default). Other values, such as "independent", suppress this part of the text.

A.2.1. Current Values: '*trust200902'

The name for these values refers to the "IETF TRUST Legal Provisions Relating to IETF Documents", sometimes simply called the "TLP, that went into effect on February 15, 2009 ([TLP2.0]). Updates to this document were published on September 12, 2009 ([TLP3.0]) and on December 28, 2009 ([TLP4.0]), modifying the license for code components (see <http://trustee.ietf.org/license-info/> for further information). The actual text is located in Section 6 ("Text To Be Included in IETF Documents") of these documents.

The tools will automatically produce the "correct" text depending on the document's date information (see above):

TLP	starting with publication date
[TLP3.0]	2009-11-01
[TLP4.0]	2010-04-01

A.2.1.1. trust200902

This should be the default, unless one of the more specific '*trust200902' values is a better fit. It produces the text in Sections 6.a and 6.b of the TLP.

A.2.1.2. noModificationTrust200902

This produces additional text from Section 6.c.i of the TLP:

> This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.
>
> Note: this clause is incompatible with RFCs that are published on the Standards Track.

A.2.1.3. noDerivativesTrust200902

This produces the additional text from Section 6.c.ii of the TLP:

> This document may not be modified, and derivative works of it may not be created, and it may not be published except as an Internet-Draft.

> Note: this clause is incompatible with RFCs.

A.2.1.4. pre5378Trust200902

This produces the additional text from Section 6.c.iii of the TLP, frequently called the "pre-5378 escape clause":

> This document may contain material from IETF Documents or IETF Contributions published or made publicly available before November 10, 2008. The person(s) controlling the copyright in some of this material may not have granted the IETF Trust the right to allow modifications of such material outside the IETF Standards Process. Without obtaining an adequate license from the person(s) controlling the copyright in such materials, this document may not be modified outside the IETF Standards Process, and derivative works of it may not be created outside the IETF Standards Process, except to format it for publication as an RFC or to translate it into languages other than English.

See Section 4 of <http://trustee.ietf.org/docs/IETF-Copyright-FAQ.pdf> for further information about when to use this value.

> Note: this text appears under "Copyright Notice", unless the document was published before November 2009, in which case it appears under "Status Of This Memo".

A.2.2. Historic Values

A.2.2.1. Historic Values: '*trust200811'

The attribute values "trust200811", "noModificationTrust200811" and "noDerivativesTrust200811" are similar to their "trust200902" counterparts, except that they use text specified in <http://trustee.ietf.org/license-info/archive/IETF-Trust-License-Policy_11-10-08.pdf>.

A.2.2.2. Historic Values: '*3978'

The attribute values "full3978", "noModification3978" and "noDerivatives3978" are similar to their counterparts above, except that they use text specified in RFC 3978 (March 2005).

A.2.2.3. Historic Values: '*3667'

The attribute values "full3667", "noModification3667" and "noDerivatives3667" are similar to their counterparts above, except that they use text specified in RFC 3667 (February 2004).

A.2.2.4. Historic Values: '*2026'

The attribute values "full2026" and "noDerivativeWorks2026" are similar to their counterparts above, except that they use text specified in RFC 2026 (October 1996).

The special value "none" was also used back then, and denied the IETF any rights beyond publication as Internet-Draft.

Appendix B. The v3 Format and Processors

This section describes topics that are specific to processors. Note that there is some discussion of processors in the main body of the document as well. For example, some elements have descriptions of how a processor might create output from the element.

A processor has multiple roles. The processor checks whether or not the input is valid XML; it also checks whether or not the input XML matches this format. It may have one or more outputs, depending on the way that it is invoked. It might convert the XML to different XML, or to a different representation such as HTML or plain text, or do no conversion at all. It also might give diagnostic warnings about how to improve the input.

In v2, the grammar was specified as a DTD. In v3, the grammar is specified only as Relax Next Generation (RNG). This means that processors need to work from the RNG, not from a DTD. Some of the features of the v3 grammar cannot be specified as a DTD.

This section differentiates between three types of processors:

- RFC Processor -- The processor used by the RFC Editor to process the input from one of the RFC streams, and to process XML produced during the RFC editing process.

- Draft Processor -- The processor used by the Internet Draft submission tool.

- Private Processors -- There may be processors that are meant to run on the computers of authors. These processors may be used to produce interim versions of the non-canonical representations so that authors can see how their XML might later be rendered; to create documents in representations different than those supported by the RFC Editor; to possibly create documents that are not meant to be Internet Drafts or RFCs; and to convert XML that has external information into XML that has that external information included.

It is expected that a single piece of software will be used for both the RFC Processor and the Draft Processor, with different application settings to differentiate between them. It is also expected that the same software could be used as a Private Processor; others might create Private Processors independently, or using parts of, the software used by the RFC Editor.

The RFC Processor and Draft Processor must have clear error reporting, giving more context than just a line number. For example, the error messages will differentiate between errors in XML and those from the v3 format.

It is expected that both the RFC Processor and the Draft Processor will have options to produce different types of output. For example, they might have a "convert XML to XML" option that expands the XML entities in the output.

B.1. Including External Text

All XML processors for v3 are expected to support XInclude [XInclude]. XInclude specifies a processing model and syntax for general purpose inclusion of information that is either on the Internet or local to the user's computer.

In the v3 syntax, XInclude is expressed as the <xi:include> element. To use this element, you need to include the "xi" namespace in the <rfc> element; that is, you need to specify

```
xmlns:xi="http://www.w3.org/2001/XInclude"
```

as one of the attributes in the <rfc> element.

The most common way to use <xi:include> is to pull in references that are already formed as XML. Currently, this can be done from xml2rfc.tools.ietf.org, but later is expected to be from the RFC Editor. For example, if a document has three normative references, all RFCs, the document might contain:

```
<references>
    <xi:include href="http://xml2rfc.tools.ietf.org/public/rfc/
       bibxml/reference.RFC.2119.xml"/>
    <xi:include href="http://xml2rfc.tools.ietf.org/public/rfc/
       bibxml/reference.RFC.4869.xml"/>
    <xi:include href="http://xml2rfc.tools.ietf.org/public/rfc/
       bibxml/reference.RFC.7169.xml"/>
</references>
```

<xi:include> can be used anywhere an XML element could be used (but

not where free text is used). For example, if three Internet Drafts are all including a particular paragraph or section verbatim, that text can be kept either in a file or somewhere on the web, and be included with <xi:include>. An example of pulling something from the local disk would be:

```
<x:include href="file://home/chris/ietf/drafts/commontext.xml"/>
```

In general, XInclude should be used instead of ENTITY references and XML Processor Instructions (PIs) that allow external inclusions.

[[anchor25: Need to think a bit more about using xi:include to replace src= for artwork or sourcecode.]]

Appendix C. Format and Content of RFCs (Future)

[[[This section describes the format of RFCs in the future. The section is in this document only temporarily: it soon move to its own page on the RFC Editor's web site.]]]

The v3 format will be used as part of the new RFC series format described in [RFC6949]. RFCs will have multiple representations: the canonical document will be in XML using the v3 format, and many non-canonical representations (such as HTML and text) will be generated as well.

C.1. Canonical RFCs

The canonical RFCs will not have any markup that uses a deprecated feature. The processor described in Appendix B will have a "convert with warnings" mode that will convert a v2 document to a v3 document that converts deprecated features wherever possible, issuing warnings for where it cannot convert. The processor will also have a "strict" mode that will issue errors if any deprecated features are in the input.

The canonical format for RFCs have a number of restrictions that the RFC Processor needs to enforce before a document can be published in its final form. In this "final" mode, the processor will give a hard error when an input XML file has:

- o any element or attribute listed as "deprecated" in the v3 vocabulary

- o XML comments

- o XML Processor Instructions

- an external DTD

- ENTITYs that have a SYSTEM component

- <artwork> elements with a "src" attribute that points to a URI with a scheme other than "data:"

- <cref> elements

The RFC Processor will have a mode that gives hard errors on all of the above other than <cref> elements. This exception is to allow the RPC to insert <cref> elements in the working XML for notes to authors in AUTH 48, notes to IANA, and so on.

The RFC Processor will also generate values for the "autogeneratedBoilerplateText" attribute in <rfc>, and "autogeneratedSectionNumber", "autogeneratedFigureNumber", and "autogeneratedTableNumber" attributes in <section>, <figure>, and <texttable> elements, respectively. These will be filled in by the RFC processor so that the canonical XML file has all of the same information as the non-canonical representations.

C.2. Non-canonical RFC Representations

As described in [RFC6949], the RFC Editor will publish non-canonical representations of RFCs. Those non-canonical representations will be derived from the canonical XML. The generation of these non-canonical representations means that there will be much more emphasis put on the RFC Processor.

[[[Stuff will go on the web page about each non-canonical representation.]]]

Appendix D. Format and Content of Internet Drafts (Future)

[[[This section describes a future where the v3 format will be allowed as the input for Internet Draft publication. The section is in this document only temporarily: it will soon move to its own page managed by someone yet to be determined.]]]

[[[Description of the transition from the current text format to also allowing v3 XML.]]]

[[[Description of what will and won't be allowed in input XML.]]]

D.1. The Draft Processor

The Draft Processor will handle input differently than the RFC Processor. For example, some XML that is valid for Internet Drafts will not be valid for canonical RFCs, so the Draft Processor can be set to give warnings for those features but the RFC Processor can be set to give hard errors.

The Draft Processor will help in the transition from the v2 format to the v3 format. A user could request that the Draft Processor convert a v2 document to v3, using the new features of v3 where appropriate, and removing the parts that are no longer supported. The processor might be able to fully convert a v2 document to v3, or it might have to only give advice on some of the v2 features that could not be automatically converted.

The Draft Processor should have a mode that mimics the RFC Processor's restictive mode, but that instead gives warnings instead of hard errors. The Draft Processor should also produce interim versions of the non-canonical representations so that authors can see how their XML might later be rendered; and to convert documents that use the v2 format to v3.

D.2. Processor Instructions

In the v2 format, XML PIs were used for a wide variety of tasks, including changing some formatting of the text outputs, adding information to HTML versions of Internet Drafts, and specifying other instructions to processors. In the v3 format, a much smaller set of PIs is supported for the Draft Processor and Private Processors. It is important to note that canonical RFCs will not contain any PIs.

The following PIs will be supported by the Draft Processor:

- editing -- When set to "yes", causes the processor to insert reference numbers throughout the output files to allow ease of referral to particular paragraphs or artwork in the text. The default is "no".

The following PIs will not be supported by the Draft Processor, but are included so that Private Processors have a common way of referring to them:

- comments -- When set to "yes", causes all <cref> elements to be rendered in the output files. The default is "yes".

- linefile -- Causes the processor to emit more detailed information about which line of which input file caused an error or warning.

Note that a Private Processor can have PIs not from this list.

Appendix E. Relax NG Schema

The following is the RelaxNG schema for the v3 format.

```
   namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"

   rfc =
     element rfc {
       attribute number { text }?,
       [ a:defaultValue = "" ] attribute obsoletes { text }?,
       [ a:defaultValue = "" ] attribute updates { text }?,
       attribute category { "std" | "bcp" | "info" | "exp" | "historic"
   }?,
       attribute consensus { "no" | "yes" }?,
       attribute seriesNo { text }?,
       attribute ipr {
         "full2026"
         | "noDerivativeWorks2026"
         | "none"
         | "full3667"
         | "noModification3667"
         | "noDerivatives3667"
         | "full3978"
         | "noModification3978"
         | "noDerivatives3978"
         | "trust200811"
         | "noModificationTrust200811"
         | "noDerivativesTrust200811"
         | "trust200902"
         | "noModificationTrust200902"
         | "noDerivativesTrust200902"
         | "pre5378Trust200902"
       }?,
       attribute iprExtract { xsd:IDREF }?,
       [ a:defaultValue = "IETF" ]
       attribute submissionType {
         "IETF" | "IAB" | "IRTF" | "independent"
       }?,
       attribute docName { text }?,
       [ a:defaultValue = "en" ] attribute xml:lang { text }?,
       [ a:defaultValue = "no" ] attribute sortRefs { "yes" | "no" }?,
       [ a:defaultValue = "yes" ] attribute symRefs { "yes" | "no" }?,
       [ a:defaultValue = "yes" ] attribute tocInclude { "yes" | "no"
   }?,
       [ a:defaultValue = "3" ] attribute tocDepth { text }?,
```

```
       attribute autogeneratedBoilerplateText { text }?,
       link?,
       front,
       middle,
       back?
     }
   link =
     element link {
       attribute href { text },
       attribute rel { text }?
     }
   front =
     element front {
       title, author+, date?, area*, workgroup*, keyword*, abstract?,
   note*
     }
   title =
     element title {
       attribute abbrev { text }?,
       attribute ascii { text }?,
       text
     }
   author =
     element author {
       attribute initials { text }?,
       attribute surname { text }?,
       attribute fullname { text }?,
       attribute role { "editor" }?,
       [ a:defaultValue = "en" ] attribute xml:lang { text }?,
       attribute ascii { text }?,
       organization?,
       address?
     }
   organization =
     element organization {
       attribute abbrev { text }?,
       attribute ascii { text }?,
       text
     }
   address =
     element address { (postal | phone | facsimile | email | uri)* }
   postal =
     element postal {
       [ a:defaultValue = "en" ] attribute xml:lang { text }?,
       ((street | city | region | code | country)* | postalLine+)
     }
   street =
     element street {
```

```
        attribute ascii { text }?,
        text
      }
    city =
      element city {
        attribute ascii { text }?,
        text
      }
    region =
      element region {
        attribute ascii { text }?,
        text
      }
    code =
      element code {
        attribute ascii { text }?,
        text
      }
    country = element country { text }
    postalLine =
      element postalLine {
        attribute ascii { text }?,
        text
      }
    phone = element phone { text }
    facsimile = element facsimile { text }
    email =
      element email {
        attribute ascii { text }?,
        text
      }
    uri = element uri { text }
    date =
      element date {
        attribute day { text }?,
        attribute month { text }?,
        attribute year { text }?,
        empty
      }
    area = element area { text }
    workgroup = element workgroup { text }
    keyword = element keyword { text }
    abstract = element abstract { t+ }
    note =
      element note {
        attribute title { text },
        t+
      }
```

```
   middle = element middle { section+ }
   section =
     element section {
       attribute anchor { xsd:ID }?,
       attribute title { text }?,
       [ a:defaultValue = "yes" ] attribute numbered { "yes" | "no" }?,
       [ a:defaultValue = "default" ]
       attribute toc { "include" | "exclude" | "default" }?,
       [ a:defaultValue = "no" ] attribute removeInRFC { "yes" | "no"
   }?,
       attribute autogeneratedSectionNumber { text }?,
       titleelement?,
       (t
        | aside
        | blockquote
        | figure
        | texttable
        | iref
        | artwork
        | sourcecode)*,
       section*
     }
   titleelement =
     element titleelement { (text | xref | eref | iref | cref | tt)* }
   t =
     element t {
       attribute anchor { xsd:ID }?,
       attribute hangText { text }?,
       attribute autogeneratedParagraphNumber { text }?,
       (text
        | \list
        | ol
        | ul
        | dl
        | figure
        | artwork
        | sourcecode
        | xref
        | eref
        | iref
        | cref
        | spanx
        | vspace
        | tt
        | strong
        | b
        | em
        | i
```

```
        | sub
        | sup)*
     }
   aside =
     element aside {
       (t | figure | texttable | iref | artwork | sourcecode)*
     }
   blockquote =
     element blockquote {
       attribute anchor { xsd:ID }?,
       attribute cite { text },
       (text
        | figure
        | artwork
        | sourcecode
        | tt
        | strong
        | b
        | em
        | i
        | sub
        | sup)*
     }
   \list =
     element list {
       [ a:defaultValue = "empty" ] attribute style { text }?,
       attribute hangIndent { text }?,
       attribute counter { text }?,
       t+
     }
   ol =
     element ol {
       [ a:defaultValue = "1" ] attribute style { text }?,
       [ a:defaultValue = "1" ] attribute start { text }?,
       attribute group { text }?,
       [ a:defaultValue = "normal" ]
       attribute spacing { "normal" | "compact" }?,
       li+
     }
   ul =
     element ul {
       [ a:defaultValue = "normal" ]
       attribute spacing { "normal" | "compact" }?,
       [ a:defaultValue = "false" ] attribute empty { "false" | "true"
   }?,
       li+
     }
   li =
```

```
     element li {
       t+
       | (text,
          ol,
          ul,
          dl,
          figure,
          artwork,
          sourcecode,
          xref,
          eref,
          iref,
          cref,
          tt,
          strong,
          b,
          em,
          i,
          sub,
          sup)+
     }
   dl =
     element dl {
       [ a:defaultValue = "normal" ]
       attribute spacing { "normal" | "compact" }?,
       [ a:defaultValue = "false" ]
       attribute hanging { "false" | "true" }?,
       (dt, dd)+
     }
   dt =
     element dt {
       (text
        | xref
        | eref
        | iref
        | cref
        | tt
        | strong
        | b
        | em
        | i
        | sub
        | sup)*
     }
   dd =
     element dd {
       t+
       | (text,
```

```
          ol,
          ul,
          dl,
          figure,
          artwork,
          sourcecode,
          xref,
          eref,
          iref,
          cref,
          tt,
          strong,
          b,
          em,
          i,
          sub,
          sup)+
     }
   xref =
     element xref {
       attribute target { xsd:IDREF },
       [ a:defaultValue = "false" ] attribute pageno { "true" | "false"
   }?,
       [ a:defaultValue = "default" ]
       attribute format { "counter" | "title" | "none" | "default" }?,
       (attribute section { text },
        attribute relative { text }?,
        attribute sectionFormat { text }?)?,
       text
     }
   eref =
     element eref {
       attribute target { text },
       text
     }
   iref =
     element iref {
       attribute item { text },
       [ a:defaultValue = "" ] attribute subitem { text }?,
       [ a:defaultValue = "false" ]
       attribute primary { "true" | "false" }?,
       empty
     }
   cref =
     element cref {
       attribute anchor { xsd:ID }?,
       attribute source { text }?,
       (text | xref | eref | tt | strong | b | em | i | sub | sup)*
```

```
      }
    tt =
      element tt {
        (text
         | xref
         | eref
         | iref
         | cref
         | strong
         | b
         | em
         | i
         | sub
         | sup)*
      }
    b =
      element b {
        (text | xref | eref | iref | cref | tt | em | i | sub | sup)*
      }
    strong =
      element strong {
        (text | xref | eref | iref | cref | tt | em | i | sub | sup)*
      }
    i =
      element i {
        (text | xref | eref | iref | cref | tt | strong | b | sub | sup)*
      }
    em =
      element em {
        (text | xref | eref | iref | cref | tt | strong | b | sub | sup)*
      }
    sub =
      element sub {
        (text | xref | eref | iref | cref | tt | strong | b | em | i)*
      }
    sup =
      element sup {
        (text | xref | eref | iref | cref | tt | strong | b | em | i)*
      }
    spanx =
      element spanx {
        [ a:defaultValue = "preserve" ]
        attribute xml:space { "default" | "preserve" }?,
        [ a:defaultValue = "emph" ] attribute style { text }?,
        text
      }
    vspace =
      element vspace {
```

```
        [ a:defaultValue = "0" ] attribute blankLines { text }?,
        empty
      }
  figure =
    element figure {
      attribute anchor { xsd:ID }?,
      [ a:defaultValue = "" ] attribute title { text }?,
      [ a:defaultValue = "false" ]
      attribute suppress-title { "true" | "false" }?,
      attribute src { text }?,
      [ a:defaultValue = "left" ]
      attribute align { "left" | "center" | "right" }?,
      [ a:defaultValue = "" ] attribute alt { text }?,
      [ a:defaultValue = "" ] attribute width { text }?,
      [ a:defaultValue = "" ] attribute height { text }?,
      attribute autogeneratedFigureNumber { text }?,
      titleelement?,
      iref*,
      preamble?,
      (artwork | sourcecode)+,
      postamble?
    }
  preamble =
    element preamble {
      (text
       | xref
       | eref
       | iref
       | cref
       | spanx
       | tt
       | strong
       | b
       | em
       | i
       | sub
       | sup)*
    }
  artwork =
    element artwork {
      attribute xml:space { text }?,
      [ a:defaultValue = "" ] attribute name { text }?,
      [ a:defaultValue = "" ] attribute type { text }?,
      attribute src { text }?,
      [ a:defaultValue = "left" ]
      attribute align { "left" | "center" | "right" }?,
      [ a:defaultValue = "" ] attribute alt { text }?,
      [ a:defaultValue = "" ] attribute width { text }?,
```

```
       [ a:defaultValue = "" ] attribute height { text }?,
       [ a:defaultValue = "en" ] attribute xml:lang { text }?,
       text*
     }
   sourcecode =
     element sourcecode {
       [ a:defaultValue = "" ] attribute name { text }?,
       [ a:defaultValue = "" ] attribute type { text }?,
       text
     }
   postamble =
     element postamble {
       (text
        | xref
        | eref
        | iref
        | cref
        | spanx
        | tt
        | strong
        | b
        | em
        | i
        | sub
        | sup)*
     }
   texttable =
     element texttable {
       attribute anchor { xsd:ID }?,
       [ a:defaultValue = "" ] attribute title { text }?,
       [ a:defaultValue = "false" ]
       attribute suppress-title { "true" | "false" }?,
       [ a:defaultValue = "center" ]
       attribute align { "left" | "center" | "right" }?,
       [ a:defaultValue = "full" ]
       attribute style { "all" | "none" | "headers" | "full" }?,
       attribute autogeneratedTableNumber { text }?,
       titleelement?,
       preamble?,
       ttcol+,
       c*,
       postamble?
     }
   ttcol =
     element ttcol {
       attribute width { text }?,
       [ a:defaultValue = "left" ]
       attribute align { "left" | "center" | "right" }?,
```

```
       (xref | eref | iref | cref | text)*
     }
   c =
     element c {
       (text
        | xref
        | eref
        | iref
        | cref
        | spanx
        | tt
        | strong
        | b
        | em
        | i
        | sub
        | sup)*
     }
   bcp14 = element bcp14 { text }
   back = element back { displayreference*, references*, section* }
   displayreference =
     element displayreference {
       attribute from { text },
       attribute to { text }
     }
   references =
     element references {
       [ a:defaultValue = "References" ] attribute title { text }?,
       reference+
     }
   reference =
     element reference {
       attribute anchor { xsd:ID },
       attribute target { text }?,
       front,
       (seriesInfo | format | refcontent | annotation)*
     }
   seriesInfo =
     element seriesInfo {
       attribute name { text },
       attribute value { text },
       attribute asciiname { text }?,
       attribute asciivalue { text }?,
       empty
     }
   format =
     element format {
       attribute target { text }?,
```

```
    attribute type { text },
    attribute octets { text }?,
    empty
  }
annotation =
  element annotation {
    attribute ascii { text }?,
    (text
     | xref
     | eref
     | iref
     | cref
     | spanx
     | tt
     | strong
     | b
     | em
     | i
     | sub
     | sup)*
  }
refcontent =
  element refcontent {
    attribute ascii { text }?,
    (text | tt | b | i | em | strong | sub | sup)*
  }
start = rfc
```

Appendix F. Schema Differences from v2

The following is a non-normative comparison of the v3 format to the v2 format. A "-" indicates lines removed from the v2 schema, and a "+" indicates lines added to the v3 schema.

```
namespace a =
"http://relaxng.org/ns/compatibility/annotations/1.0"

rfc =
  element rfc {
    attribute number { text }?,
    [ a:defaultValue = "" ] attribute obsoletes { text }?,
    [ a:defaultValue = "" ] attribute updates { text }?,
    attribute category { "std" | "bcp" | "info" | "exp" |
"historic" }?,
    attribute consensus { "no" | "yes" }?,
    attribute seriesNo { text }?,
```

```
         attribute ipr {
           "full2026"
           | "noDerivativeWorks2026"
           | "none"
           | "full3667"
           | "noModification3667"
           | "noDerivatives3667"
           | "full3978"
           | "noModification3978"
           | "noDerivatives3978"
           | "trust200811"
           | "noModificationTrust200811"
           | "noDerivativesTrust200811"
           | "trust200902"
           | "noModificationTrust200902"
           | "noDerivativesTrust200902"
           | "pre5378Trust200902"
         }?,
         attribute iprExtract { xsd:IDREF }?,
         [ a:defaultValue = "IETF" ]
         attribute submissionType {
           "IETF" | "IAB" | "IRTF" | "independent"
         }?,
         attribute docName { text }?,
         [ a:defaultValue = "en" ] attribute xml:lang { text }?,
   +     [ a:defaultValue = "no" ] attribute sortRefs { "yes" | "no"
   + }?,
   +     [ a:defaultValue = "yes" ] attribute symRefs { "yes" | "no"
   + }?,
   +     [ a:defaultValue = "yes" ] attribute tocInclude { "yes" |
   + "no" }?,
   +     [ a:defaultValue = "3" ] attribute tocDepth { text }?,
   +     attribute autogeneratedBoilerplateText { text }?,
   +     link?,
         front,
         middle,
         back?
       }
   + link =
   +   element link {
   +     attribute href { text },
   +     attribute rel { text }?
   +   }
     front =
       element front {
   -     title, author+, date, area*, workgroup*, keyword*, abstract?,
   - note*
   +     title, author+, date?, area*, workgroup*, keyword*,
```

```
   + abstract?, note*
       }
     title =
       element title {
         attribute abbrev { text }?,
   +     attribute ascii { text }?,
         text
       }
     author =
       element author {
         attribute initials { text }?,
         attribute surname { text }?,
         attribute fullname { text }?,
         attribute role { "editor" }?,
   +     [ a:defaultValue = "en" ] attribute xml:lang { text }?,
   +     attribute ascii { text }?,
         organization?,
         address?
       }
     organization =
       element organization {
         attribute abbrev { text }?,
   +     attribute ascii { text }?,
   +     text
   +   }
   + address =
   +   element address { (postal | phone | facsimile | email | uri)* }
   + postal =
   +   element postal {
   +     [ a:defaultValue = "en" ] attribute xml:lang { text }?,
   +     ((street | city | region | code | country)* | postalLine+)
   +   }
   + street =
   +   element street {
   +     attribute ascii { text }?,
   +     text
   +   }
   + city =
   +   element city {
   +     attribute ascii { text }?,
   +     text
   +   }
   + region =
   +   element region {
   +     attribute ascii { text }?,
   +     text
   +   }
   + code =
```

```
   +   element code {
   +     attribute ascii { text }?,
         text
       }
   - address = element address { postal?, phone?, facsimile?, email?,
   - uri? }
   - postal = element postal { street+, (city | region | code |
   - country)* }
   - street = element street { text }
   - city = element city { text }
   - region = element region { text }
   - code = element code { text }
     country = element country { text }
   + postalLine =
   +   element postalLine {
   +     attribute ascii { text }?,
   +     text
   +   }
     phone = element phone { text }
     facsimile = element facsimile { text }
   - email = element email { text }
   + email =
   +   element email {
   +     attribute ascii { text }?,
   +     text
   +   }
     uri = element uri { text }
     date =
       element date {
         attribute day { text }?,
         attribute month { text }?,
         attribute year { text }?,
         empty
       }
     area = element area { text }
     workgroup = element workgroup { text }
     keyword = element keyword { text }
     abstract = element abstract { t+ }
     note =
       element note {
         attribute title { text },
         t+
       }
     middle = element middle { section+ }
     section =
       element section {
         attribute anchor { xsd:ID }?,
   -     attribute title { text },
```

```
   +     attribute title { text }?,
   +     [ a:defaultValue = "yes" ] attribute numbered { "yes" | "no"
   + }?,
         [ a:defaultValue = "default" ]
         attribute toc { "include" | "exclude" | "default" }?,
   -     (t | figure | texttable | iref)*,
   +     [ a:defaultValue = "no" ] attribute removeInRFC { "yes" |
   + "no" }?,
   +     attribute autogeneratedSectionNumber { text }?,
   +     titleelement?,
   +     (t
   +      | aside
   +      | blockquote
   +      | figure
   +      | texttable
   +      | iref
   +      | artwork
   +      | sourcecode)*,
         section*
       }
   + titleelement =
   +   element titleelement { (text | xref | eref | iref | cref | tt)*
   + }
     t =
       element t {
         attribute anchor { xsd:ID }?,
         attribute hangText { text }?,
   +     attribute autogeneratedParagraphNumber { text }?,
         (text
          | \list
   +      | ol
   +      | ul
   +      | dl
          | figure
   +      | artwork
   +      | sourcecode
          | xref
          | eref
          | iref
          | cref
          | spanx
   -      | vspace)*
   +      | vspace
   +      | tt
   +      | strong
   +      | b
   +      | em
   +      | i
```

```
   +      | sub
   +      | sup)*
   +   }
   + aside =
   +   element aside {
   +     (t | figure | texttable | iref | artwork | sourcecode)*
   +   }
   + blockquote =
   +   element blockquote {
   +     attribute anchor { xsd:ID }?,
   +     attribute cite { text },
   +     (text
   +      | figure
   +      | artwork
   +      | sourcecode
   +      | tt
   +      | strong
   +      | b
   +      | em
   +      | i
   +      | sub
   +      | sup)*
       }
     \list =
       element list {
   -     attribute style { text }?,
   +     [ a:defaultValue = "empty" ] attribute style { text }?,
         attribute hangIndent { text }?,
         attribute counter { text }?,
         t+
       }
   + ol =
   +   element ol {
   +     [ a:defaultValue = "1" ] attribute style { text }?,
   +     [ a:defaultValue = "1" ] attribute start { text }?,
   +     attribute group { text }?,
   +     [ a:defaultValue = "normal" ]
   +     attribute spacing { "normal" | "compact" }?,
   +     li+
   +   }
   + ul =
   +   element ul {
   +     [ a:defaultValue = "normal" ]
   +     attribute spacing { "normal" | "compact" }?,
   +     [ a:defaultValue = "false" ] attribute empty { "false" |
   + "true" }?,
   +     li+
   +   }
```

```
   + li =
   +   element li {
   +     t+
   +     | (text,
   +        ol,
   +        ul,
   +        dl,
   +        figure,
   +        artwork,
   +        sourcecode,
   +        xref,
   +        eref,
   +        iref,
   +        cref,
   +        tt,
   +        strong,
   +        b,
   +        em,
   +        i,
   +        sub,
   +        sup)+
   +   }
   + dl =
   +   element dl {
   +     [ a:defaultValue = "normal" ]
   +     attribute spacing { "normal" | "compact" }?,
   +     [ a:defaultValue = "false" ]
   +     attribute hanging { "false" | "true" }?,
   +     (dt, dd)+
   +   }
   + dt =
   +   element dt {
   +     (text
   +      | xref
   +      | eref
   +      | iref
   +      | cref
   +      | tt
   +      | strong
   +      | b
   +      | em
   +      | i
   +      | sub
   +      | sup)*
   +   }
   + dd =
   +   element dd {
   +     t+
```

```
   +     | (text,
   +        ol,
   +        ul,
   +        dl,
   +        figure,
   +        artwork,
   +        sourcecode,
   +        xref,
   +        eref,
   +        iref,
   +        cref,
   +        tt,
   +        strong,
   +        b,
   +        em,
   +        i,
   +        sub,
   +        sup)+
   +   }
     xref =
       element xref {
         attribute target { xsd:IDREF },
         [ a:defaultValue = "false" ] attribute pageno { "true" |
     "false" }?,
         [ a:defaultValue = "default" ]
         attribute format { "counter" | "title" | "none" | "default"
     }?,
   +     (attribute section { text },
   +      attribute relative { text }?,
   +      attribute sectionFormat { text }?)?,
         text
       }
     eref =
       element eref {
         attribute target { text },
         text
       }
     iref =
       element iref {
         attribute item { text },
         [ a:defaultValue = "" ] attribute subitem { text }?,
         [ a:defaultValue = "false" ]
         attribute primary { "true" | "false" }?,
         empty
       }
     cref =
       element cref {
         attribute anchor { xsd:ID }?,
```

```
          attribute source { text }?,
   -      text
   +      (text | xref | eref | tt | strong | b | em | i | sub | sup)*
   +    }
   + tt =
   +    element tt {
   +      (text
   +       | xref
   +       | eref
   +       | iref
   +       | cref
   +       | strong
   +       | b
   +       | em
   +       | i
   +       | sub
   +       | sup)*
   +    }
   + b =
   +    element b {
   +      (text | xref | eref | iref | cref | tt | em | i | sub | sup)*
   +    }
   + strong =
   +    element strong {
   +      (text | xref | eref | iref | cref | tt | em | i | sub | sup)*
   +    }
   + i =
   +    element i {
   +      (text | xref | eref | iref | cref | tt | strong | b | sub |
   + sup)*
   +    }
   + em =
   +    element em {
   +      (text | xref | eref | iref | cref | tt | strong | b | sub |
   + sup)*
   +    }
   + sub =
   +    element sub {
   +      (text | xref | eref | iref | cref | tt | strong | b | em | i)*
   +    }
   + sup =
   +    element sup {
   +      (text | xref | eref | iref | cref | tt | strong | b | em | i)*
        }
     spanx =
        element spanx {
          [ a:defaultValue = "preserve" ]
          attribute xml:space { "default" | "preserve" }?,
```

```
         [ a:defaultValue = "emph" ] attribute style { text }?,
         text
       }
     vspace =
       element vspace {
         [ a:defaultValue = "0" ] attribute blankLines { text }?,
         empty
       }
     figure =
       element figure {
         attribute anchor { xsd:ID }?,
         [ a:defaultValue = "" ] attribute title { text }?,
         [ a:defaultValue = "false" ]
         attribute suppress-title { "true" | "false" }?,
         attribute src { text }?,
         [ a:defaultValue = "left" ]
         attribute align { "left" | "center" | "right" }?,
         [ a:defaultValue = "" ] attribute alt { text }?,
         [ a:defaultValue = "" ] attribute width { text }?,
         [ a:defaultValue = "" ] attribute height { text }?,
   +     attribute autogeneratedFigureNumber { text }?,
   +     titleelement?,
         iref*,
         preamble?,
   -     artwork,
   +     (artwork | sourcecode)+,
         postamble?
       }
     preamble =
   -   element preamble { (text | xref | eref | iref | cref | spanx)* }
   +   element preamble {
   +     (text
   +      | xref
   +      | eref
   +      | iref
   +      | cref
   +      | spanx
   +      | tt
   +      | strong
   +      | b
   +      | em
   +      | i
   +      | sub
   +      | sup)*
   +   }
     artwork =
       element artwork {
   -     [ a:defaultValue = "preserve" ]
```

```
   -     attribute xml:space { "default" | "preserve" }?,
   +     attribute xml:space { text }?,
         [ a:defaultValue = "" ] attribute name { text }?,
         [ a:defaultValue = "" ] attribute type { text }?,
         attribute src { text }?,
         [ a:defaultValue = "left" ]
         attribute align { "left" | "center" | "right" }?,
         [ a:defaultValue = "" ] attribute alt { text }?,
         [ a:defaultValue = "" ] attribute width { text }?,
         [ a:defaultValue = "" ] attribute height { text }?,
   +     [ a:defaultValue = "en" ] attribute xml:lang { text }?,
         text*
       }
   + sourcecode =
   +   element sourcecode {
   +     [ a:defaultValue = "" ] attribute name { text }?,
   +     [ a:defaultValue = "" ] attribute type { text }?,
   +     text
   +   }
     postamble =
   -   element postamble { (text | xref | eref | iref | cref | spanx)*
   +   element postamble {
   +     (text
   +      | xref
   +      | eref
   +      | iref
   +      | cref
   +      | spanx
   +      | tt
   +      | strong
   +      | b
   +      | em
   +      | i
   +      | sub
   +      | sup)*
     }
     texttable =
       element texttable {
         attribute anchor { xsd:ID }?,
         [ a:defaultValue = "" ] attribute title { text }?,
         [ a:defaultValue = "false" ]
         attribute suppress-title { "true" | "false" }?,
         [ a:defaultValue = "center" ]
         attribute align { "left" | "center" | "right" }?,
         [ a:defaultValue = "full" ]
         attribute style { "all" | "none" | "headers" | "full" }?,
   +     attribute autogeneratedTableNumber { text }?,
   +     titleelement?,
```

```
             preamble?,
             ttcol+,
             c*,
             postamble?
           }
         ttcol =
           element ttcol {
             attribute width { text }?,
             [ a:defaultValue = "left" ]
             attribute align { "left" | "center" | "right" }?,
       -     text
       +     (xref | eref | iref | cref | text)*
       +   }
       + c =
       +   element c {
       +     (text
       +      | xref
       +      | eref
       +      | iref
       +      | cref
       +      | spanx
       +      | tt
       +      | strong
       +      | b
       +      | em
       +      | i
       +      | sub
       +      | sup)*
       +   }
       + bcp14 = element bcp14 { text }
       + back = element back { displayreference*, references*, section* }
       + displayreference =
       +   element displayreference {
       +     attribute from { text },
       +     attribute to { text }
           }
       - c = element c { (text | xref | eref | iref | cref | spanx)* }
       - back = element back { references*, section* }
         references =
           element references {
             [ a:defaultValue = "References" ] attribute title { text }?,
             reference+
           }
         reference =
           element reference {
             attribute anchor { xsd:ID },
             attribute target { text }?,
             front,
```

```
   -     seriesInfo*,
   -     format*,
   -     annotation*
   +     (seriesInfo | format | refcontent | annotation)*
       }
     seriesInfo =
       element seriesInfo {
         attribute name { text },
         attribute value { text },
   +     attribute asciiname { text }?,
   +     attribute asciivalue { text }?,
         empty
       }
     format =
       element format {
         attribute target { text }?,
         attribute type { text },
         attribute octets { text }?,
         empty
       }
     annotation =
   -   element annotation { (text | xref | eref | iref | cref |
   - spanx)* }
   +   element annotation {
   +     attribute ascii { text }?,
   +     (text
   +      | xref
   +      | eref
   +      | iref
   +      | cref
   +      | spanx
   +      | tt
   +      | strong
   +      | b
   +      | em
   +      | i
   +      | sub
   +      | sup)*
   +   }
   + refcontent =
   +   element refcontent {
   +     attribute ascii { text }?,
   +     (text | tt | b | i | em | strong | sub | sup)*
   +   }
     start = rfc
```

Index

Author's Address

Paul Hoffman
VPN Consortium

EMail: paul.hoffman@vpnc.org