

Network Working Group
Internet-Draft
Obsoletes: [2629](#) (if approved)
Intended status: Standards Track
Expires: October 21, 2014

P. Hoffman
VPN Consortium
April 19, 2014

**The 'XML2RFC' version 3 Vocabulary
draft-hoffman-xml2rfc-05**

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.

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>](https://www.rfc-editor.org/mailman/listinfo/rfc-interest).

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

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

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

This Internet-Draft will expire on October 21, 2014.

Copyright Notice

Copyright (c) 2014 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of

publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	5
1.1.	Design Criteria for the Changes in v3	5
1.2.	Differences from v2 to v3	6
1.2.1.	New Elements in v3	6
1.2.2.	New Attributes for Existing Elements	7
1.2.3.	Elements and Attributes Deprecated from v2	7
1.2.4.	Additional Changes from v2	8
1.3.	Syntax Notation	9
2.	Elements	9
2.1.	<abstract>	9
2.2.	<address>	9
2.3.	<annotation>	10
2.4.	<area>	11
2.5.	<artwork>	11
2.6.	<aside>	13
2.7.	<author>	14
2.8.		15
2.9.	<back>	16
2.10.	<blockquote>	16
2.11.	<c>	17
2.12.	<city>	18
2.13.	<code>	18
2.14.	<country>	18
2.15.	<cref>	18
2.16.	<date>	19
2.17.	<dd>	20
2.18.	<displayreference>	21
2.19.	<dl>	21
2.20.	<dt>	22
2.21.		23
2.22.	<email>	24
2.23.	<eref>	24
2.24.	<facsimile>	24
2.25.	<figure>	25
2.26.	<format>	27
2.27.	<front>	27
2.28.	<i>	28
2.29.	<ieref>	28
2.30.	<keyword>	29

Hoffman

Expires October 21, 2014

[Page 2]

2.31.		29
2.32.	<list>	30
2.33.	<middle>	31
2.34.	<note>	31
2.35.		32
2.36.	<organization>	33
2.37.	<phone>	33
2.38.	<postal>	34
2.39.	<postalLine>	34
2.40.	<postamble>	35
2.41.	<preamble>	35
2.42.	<refcontent>	36
2.43.	<reference>	37
2.44.	<references>	38
2.45.	<region>	38
2.46.	<rfc>	38
2.47.	<section>	43
2.48.	<seriesInfo>	45
2.49.	<sourcecode>	45
2.50.	<spanx>	47
2.51.	<street>	47
2.52.		48
2.53.	<t>	48
2.54.	<texttable>	49
2.55.	<title>	51
2.56.	<titleelement>	52
2.57.	<tt>	52
2.58.	<ttcol>	53
2.59.		54
2.60.	<uri>	54
2.61.	<vspace>	55
2.62.	<workgroup>	55
2.63.	<xref>	55
3.	Special Unicode Code Points	58
4.	Internationalization Considerations	59
5.	Security Considerations	59
6.	IANA Considerations	60
6.1.	Internet Media Type Registration	60
7.	Acknowledgments	61
8.	References	61
8.1.	Normative References	61
8.2.	Informative References	61
Appendix A.	Front Page Generation	63
A.1.	The /rfc/@category Attribute	63
A.2.	The /rfc/@ipr Attribute	64
A.2.1.	Current Values: '*trust200902'	64
A.2.2.	Historic Values	66
Appendix B.	The v3 Format and Processors	66

Hoffman

Expires October 21, 2014

[Page 3]

B.1.	Including External Text	68
B.2.	The RFC Processor	69
B.3.	The Draft Processor	69
B.4.	Processor Instructions	70
Appendix C.	Relax NG Schema	70
Appendix D.	Schema Differences from v2	80
Index	90

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 series will have multiple formats: the canonical document will be in XML using the v3 format described here, and many non-canonical formats (such as HTML and text) will be generated as well. The generation of these non-canonical formats means that there will be much more emphasis put on the processor that handles the XML. Features of the expected processor are described in [Appendix B](#).

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

The following is a hopefully-complete list of all the technical changes between [[XML2RFCv2](#)] and this document, as well as the design criteria for those changes. Note that the list is for the current version of this document only. There are additional changes that are expected to the v3 vocabulary that are being discussed. Also note that changes to the design choices for the differences are also expected.

1.1. Design Criteria for the Changes in v3

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

- o The intention is that starting and editing a v3 document will be easier than for a v2 document.
- o There will be good v2-to-v3 conversion tools for when an author wants to change versions.
- o There are no current plans to make v3 XML the required submission format 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 described in [[XML2RFCv2](#)].

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.

[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

- o Add <dl>, , and as new ways to make lists. This is a significant change from v2 in that the child under these elements is , not <t>. has a model of either containing one or more <t> elements, or containing the flowing text normally found in <t>.
- o Add , , , <i>, and <tt> for character formatting.
- o Add <aside> for incidental text that will be indented when displayed.
- o Add <sourcecode> to differentiate from <artwork>.
- o Add <blockquote> to indicate a quotation as in a paragraph-like format.
- o Add <titleelement> to sections, figures, and texttables to allow character formatting (fixed-width font) in their titles, and to allow references.

- o Add `<postalline>`, free text that represents one line of the address.
- o Add `<displayreference>` to allow display of more mnemonic anchor names for automatically-included references.
- o Add `<refcontent>` to allow better control of text in a reference.

1.2.2. New Attributes for Existing Elements

- o 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.
- o Add "ascii" attributes to `<author>`, `<email>`, and `<organization>`. 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.
- o 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.
- o Add the "section", "relative", and "sectionFormat" attributes to `<xref>`.
- o Add the "numbered" and "removeinrfc" attributes to `<section>`.
- o Add "autogeneratedBoilerplateText", "autogeneratedSectionNumber", "autogeneratedFigureNumber", and "autogeneratedTableNumber" attributes to `<rfc>`, `<section>`, `<figure>`, and `<texttable>`, 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 formats.

1.2.3. Elements and Attributes Deprecated from v2

- o Deprecate `<list>` in favor of `<dl>`, ``, and ``.
- o Deprecate `<spanx>`; replace it with ``, ``, ``, `<i>`, and `<tt>`.
- o Deprecate `<vspace>` because the major use for it, creating pseudo-paragraph-breaks in lists, is now handled properly.
- o Deprecate `<facsimile>` because it is rarely used and is not actually useful.

- o Deprecate `<format>` because it is not useful and has caused surprise for authors in the past.
- o Deprecate the "title" attribute in `<section>`, `<figure>`, and `<texttable>` in favor of the new `<titleelement>`.
- o Deprecate the "alt", "height", "src", and "width" attributes in `<figure>` because they overlap with the attributes in `<artwork>`.
- o Deprecate the "xml:space" attribute in `<artwork>` because there was only one useful value. Deprecate "height" and "width" attribute.
- o 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

- o Allow `<artwork>` to be used on its own in `<section>` (no longer confine it to a figure).
- o Give more specifics of handling the "type" attribute in `<artwork>`.
- o In `<address>`, allow the sub-elements to be in any order.
- o Allow ``, ``, ``, `<i>`, `<tt>`, `<eref>`, and `<xref>` in `<cref>`.
- o Allow the sub-elements inside a `<reference>` to be in any order.
- o Turned off the auto-generation of anchors in `<cref>` because there is no use case for them that cannot be achieved in other ways.
- o Allow more than one `<artwork>`, or more than one `<sourcecode>`, in `<figure>`.
- o In `<front>`, make `<date>` optional.
- o In `<postal>`, allow the sub-elements to be in any order. Also allow the inclusion of the new `<postalLine>` instead of the older elements.
- o In `<ttcol>`, allow `<xref>`, `<eref>`, `<ieref>`, and `<cref>` as optional children.
- o Do not generate the grammar from a DTD, but instead get it directly from the Relax Next Generation (RNG) grammar.

1.3. Syntax Notation

The XML vocabulary here is defined in prose, based on the Relax NG schema ([[RNC](#)]) contained in [Appendix C](#) (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 2.27](#)).

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:

- o <postal> elements ([Section 2.38](#))
- o <phone> elements ([Section 2.37](#))
- o <facsimile> elements ([Section 2.24](#))

- o <email> elements ([Section 2.22](#))
- o <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 2.43](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <spanx> elements ([Section 2.50](#))
- o <tt> elements ([Section 2.57](#))
- o elements ([Section 2.52](#))
- o elements ([Section 2.8](#))
- o elements ([Section 2.21](#))
- o <i> elements ([Section 2.28](#))

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>](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", but that list will likely change over time.

This element appears as child element of: <front> ([Section 2.27](#)).

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:

- o diagrams ("line art"),
- o complex tables, or
- o 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 formats 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, those characters either need to be escaped or the artwork needs to be in a CDATA structure; 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>.

This element appears as child element of: <aside> ([Section 2.6](#)), <blockquote> ([Section 2.10](#)), <dd> ([Section 2.17](#)), <figure> ([Section 2.25](#)), ([Section 2.31](#)), <section> ([Section 2.47](#)), 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:

- o "left" (default)
- o "center"
- o "right"

[2.5.2.](#) **'alt' attribute**

Alternative text description of the artwork (not just the caption).

[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](#)) as well, in which case the graphics file is wholly part of the XML file.

[[anchor8: Add an example of in-lined SVG here.]]

[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 in the non-canonical output formats. 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:

- o ascii-art
- o call-flow
- o hex-dump
- o 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 2.47](#)).

Content model:

In any order:

- o `<t>` elements ([Section 2.53](#))
- o `<figure>` elements ([Section 2.25](#))
- o `<texttable>` elements ([Section 2.54](#))
- o `<iref>` elements ([Section 2.29](#))
- o `<artwork>` elements ([Section 2.5](#))
- o `<sourcecode>` elements ([Section 2.49](#))

[2.7.](#) `<author>`

Provides information about a document author.

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". For RFCs, it is the stream that determines who is listed as authors or editors.

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

This element appears as child element of: `<front>` ([Section 2.27](#)).

Content model:

In this order:

1. One optional `<organization>` element ([Section 2.36](#))
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.

[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.](#)

Causes the text to be displayed in bold. It is almost always a better idea to use the element instead.

This element appears as child element of: <annotation> ([Section 2.3](#)), <blockquote> ([Section 2.10](#)), <c> ([Section 2.11](#)), <cref> ([Section 2.15](#)), <dd> ([Section 2.17](#)), <dt> ([Section 2.20](#)), ([Section 2.21](#)), <i> ([Section 2.28](#)), ([Section 2.31](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), <refcontent> ([Section 2.42](#)), <t> ([Section 2.53](#)), and <tt> ([Section 2.57](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <tt> elements ([Section 2.57](#))
- o elements ([Section 2.21](#))
- o <i> elements ([Section 2.28](#))

[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 2.46](#)).

Content model:

In this order:

1. Optional <displayreference> elements ([Section 2.18](#))
2. Optional <references> elements ([Section 2.44](#))
3. Optional <section> elements ([Section 2.47](#))

[2.10.](#) <blockquote>

Specifies a block of text is a quotation. The "cite" attribute is required, and must be a URI.

Content model:

In any order:

- o Text
- o <figure> elements ([Section 2.25](#))
- o <artwork> elements ([Section 2.5](#))

- o `<sourcecode>` elements ([Section 2.49](#))
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.52](#))
- o `` elements ([Section 2.8](#))
- o `` elements ([Section 2.21](#))
- o `<i>` elements ([Section 2.28](#))

[2.10.1.](#) 'anchor' attribute

Document-wide unique identifier for this quotation.

[2.10.2.](#) 'cite' attribute (mandatory)

The source of the citation. This must be a URI. [[anchor9: Needs an example of a cite for a reference that is already in the spec.]]

[2.11.](#) `<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:

- o Text
- o `<xref>` elements ([Section 2.63](#))
- o `<eref>` elements ([Section 2.23](#))
- o `<iref>` elements ([Section 2.29](#))
- o `<cref>` elements ([Section 2.15](#))
- o `<spanx>` elements ([Section 2.50](#))
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.52](#))

- o `` elements ([Section 2.8](#))
- o `` elements ([Section 2.21](#))
- o `<i>` elements ([Section 2.28](#))

[2.12.](#) `<city>`

Gives the city name in a postal address.

This element appears as child element of: `<postal>` ([Section 2.38](#)).

Content model: only text content.

[2.13.](#) `<code>`

Gives the postal region code.

This element appears as child element of: `<postal>` ([Section 2.38](#)).

Content model: only text content.

[2.14.](#) `<country>`

Gives the country in a postal address.

This element appears as child element of: `<postal>` ([Section 2.38](#)).

Content model: only text content.

[2.15.](#) `<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 format and settings of the formatter), or not at all (when generating an RFC).

This element appears as child element of: `<annotation>` ([Section 2.3](#)), `` ([Section 2.8](#)), `<c>` ([Section 2.11](#)), `<dd>` ([Section 2.17](#)), `<dt>` ([Section 2.20](#)), `` ([Section 2.21](#)), `<i>` ([Section 2.28](#)), `` ([Section 2.31](#)), `<postamble>` ([Section 2.40](#)), `<preamble>` ([Section 2.41](#)), `` ([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:

- o Text
- o `<xref>` elements ([Section 2.63](#))
- o `<eref>` elements ([Section 2.23](#))
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.52](#))
- o `` elements ([Section 2.8](#))
- o `` elements ([Section 2.21](#))
- o `<i>` elements ([Section 2.28](#))

[2.15.1.](#) 'anchor' attribute

Document-wide unique identifier for this comment.

[2.15.2.](#) 'source' attribute

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

[2.16.](#) `<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 first case, it defines the publication date, which, when producing Internet-Drafts, will be used for computing the expiration date (see [[IDGUIDE](#)]). When "year", "month" or "day" are left out, the processor will attempt to use the current system date if the attributes that are specified do match the system date.

Note 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 second case, the date information will be embedded as-is into the reference text. Therefore, also vague dates ("ca. 2000"), date ranges, and so on, are allowed.

This element appears as child element of: <front> ([Section 2.27](#)).

Content model: this element does not have any contents.

[2.16.1.](#) 'day' attribute

Day of publication; this is a number.

[2.16.2.](#) 'month' attribute

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

[2.16.3.](#) 'year' attribute

Year of publication.

[2.17.](#) <dd>

The term being defined in a definition list.

This element appears as child element of: <dl> ([Section 2.19](#)).

Content model:

Either:

- o One or more <t> elements ([Section 2.53](#))

Or:

- o Text

One or more elements ([Section 2.35](#))

One or more elements ([Section 2.59](#))

One or more <dl> elements ([Section 2.19](#))

One or more <figure> elements ([Section 2.25](#))

One or more <artwork> elements ([Section 2.5](#))

One or more <sourcecode> elements ([Section 2.49](#))

One or more <xref> elements ([Section 2.63](#))

One or more <eref> elements ([Section 2.23](#))

One or more <iref> elements ([Section 2.29](#))

One or more <cref> elements ([Section 2.15](#))

One or more <tt> elements ([Section 2.57](#))

One or more elements ([Section 2.52](#))

One or more elements ([Section 2.8](#))

One or more elements ([Section 2.21](#))

One or more <i> elements ([Section 2.28](#))

[2.18.](#) <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 mnemonic 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.

[2.18.1.](#) 'from' attribute (mandatory)

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

[2.18.2.](#) '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 "_".

[2.19.](#) <dl>

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

This element appears as child element of: <dd> ([Section 2.17](#)),

([Section 2.31](#)), and <t> ([Section 2.53](#)).

Content model:

One or more <dt> elements ([Section 2.20](#))

One or more <dd> elements ([Section 2.17](#))

[2.19.1.](#) '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:

- o "false" (default)
- o "true"

[2.19.2.](#) '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"

[2.20.](#) <dt>

The definition part of an entry in a definition list.

This element appears as child element of: <dl> ([Section 2.19](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))

- o `<eref>` elements ([Section 2.23](#))
- o `<ieref>` elements ([Section 2.29](#))
- o `<cref>` elements ([Section 2.15](#))
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.52](#))
- o `` elements ([Section 2.8](#))
- o `` elements ([Section 2.21](#))
- o `<i>` elements ([Section 2.28](#))

[2.21](#). ``

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](#)), `` ([Section 2.8](#)), `<blockquote>` ([Section 2.10](#)), `<c>` ([Section 2.11](#)), `<cref>` ([Section 2.15](#)), `<dd>` ([Section 2.17](#)), `<dt>` ([Section 2.20](#)), `` ([Section 2.31](#)), `<postamble>` ([Section 2.40](#)), `<preamble>` ([Section 2.41](#)), `<refcontent>` ([Section 2.42](#)), `` ([Section 2.52](#)), `<t>` ([Section 2.53](#)), and `<tt>` ([Section 2.57](#)).

Content model:

In any order:

- o Text
- o `<xref>` elements ([Section 2.63](#))
- o `<eref>` elements ([Section 2.23](#))
- o `<ieref>` elements ([Section 2.29](#))
- o `<cref>` elements ([Section 2.15](#))
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.52](#))

- o `` elements ([Section 2.8](#))

2.22. `<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.

2.22.1. '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.

2.23. `<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](#)), `` ([Section 2.8](#)), `<c>` ([Section 2.11](#)), `<cref>` ([Section 2.15](#)), `<dd>` ([Section 2.17](#)), `<dt>` ([Section 2.20](#)), `` ([Section 2.21](#)), `<i>` ([Section 2.28](#)), `` ([Section 2.31](#)), `<postamble>` ([Section 2.40](#)), `<preamble>` ([Section 2.41](#)), `` ([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.

2.23.1. 'target' attribute (mandatory)

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

2.24. `<facsimile>`

Deprecated.

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

Content model: only text content.

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 2.10](#)), <dd> ([Section 2.17](#)), ([Section 2.31](#)), <section> ([Section 2.47](#)), and <t> ([Section 2.53](#)).

Content model:

In this order:

1. One optional <titleelement> element ([Section 2.56](#))
2. Optional <iref> elements ([Section 2.29](#))
3. One optional <preamble> element ([Section 2.41](#))
4. Either:
 - * One or more <artwork> elements ([Section 2.5](#))
5. Or:
 - * One or more <sourcecode> elements ([Section 2.49](#))
6. One optional <postamble> element ([Section 2.40](#))

2.25.1. 'align' attribute

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

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

Allowed values:

- o "left" (default)
- o "center"
- o "right"

[2.25.2.](#) **'alt' attribute**

Deprecated.

[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 is absent. Setting this attribute to "true" will prevent this.

Allowed values:

- o "true"
- o "false" (default)

[2.25.8.](#) **'title' attribute**

Deprecated. Use <titleelement> instead.

[2.25.9.](#) **'width' attribute**

Deprecated.

[2.26.](#) <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 2.43](#)).

Content model: this element does not have any contents.

[2.26.1.](#) 'octets' attribute

Deprecated.

[2.26.2.](#) 'target' attribute

Deprecated.

[2.26.3.](#) 'type' attribute (mandatory)

Deprecated.

[2.27.](#) <front>

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

This element appears as child element of: <reference> ([Section 2.43](#)), and <rfc> ([Section 2.46](#)).

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 2.16](#))
4. Optional <area> elements ([Section 2.4](#))
5. Optional <workgroup> elements ([Section 2.62](#))
6. Optional <keyword> elements ([Section 2.30](#))
7. One optional <abstract> element ([Section 2.1](#))

8. Optional <note> elements ([Section 2.34](#))

[2.28.](#) <i>

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

This element appears as child element of: <annotation> ([Section 2.3](#)), ([Section 2.8](#)), <blockquote> ([Section 2.10](#)), <c> ([Section 2.11](#)), <cref> ([Section 2.15](#)), <dd> ([Section 2.17](#)), <dt> ([Section 2.20](#)), ([Section 2.31](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), <refcontent> ([Section 2.42](#)), ([Section 2.52](#)), <t> ([Section 2.53](#)), and <tt> ([Section 2.57](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <tt> elements ([Section 2.57](#))
- o elements ([Section 2.52](#))
- o elements ([Section 2.8](#))

[2.29.](#) <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](#)), ([Section 2.8](#)), <c> ([Section 2.11](#)), <dd> ([Section 2.17](#)), <dt> ([Section 2.20](#)), ([Section 2.21](#)), <figure> ([Section 2.25](#)), <i> ([Section 2.28](#)), ([Section 2.31](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), <section> ([Section 2.47](#)), ([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.

[2.29.1.](#) 'item' attribute (mandatory)

The item to include.

[2.29.2.](#) 'primary' attribute

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

Allowed values:

- o "true"
- o "false" (default)

[2.29.3.](#) 'subitem' attribute

The subitem to include.

[2.30.](#) <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 formats.

This element appears as child element of: <front> ([Section 2.27](#)).

Content model: only text content.

[2.31.](#)

A list element, used in and .

This element appears as child element of: ([Section 2.35](#)), and

 ([Section 2.59](#)).

Content model:

Either:

- o One or more <t> elements ([Section 2.53](#))

Or:

- o Text

- One or more elements ([Section 2.35](#))

- One or more elements ([Section 2.59](#))

- One or more <dl> elements ([Section 2.19](#))

- One or more <figure> elements ([Section 2.25](#))

- One or more <artwork> elements ([Section 2.5](#))

- One or more <sourcecode> elements ([Section 2.49](#))

- One or more <xref> elements ([Section 2.63](#))

- One or more <eref> elements ([Section 2.23](#))

- One or more <iref> elements ([Section 2.29](#))

- One or more <cref> elements ([Section 2.15](#))

- One or more <tt> elements ([Section 2.57](#))

- One or more elements ([Section 2.52](#))

- One or more elements ([Section 2.8](#))

- One or more elements ([Section 2.21](#))

- One or more <i> elements ([Section 2.28](#))

[2.32.](#) <list>

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

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

Content model:

One or more <t> elements ([Section 2.53](#))

[2.32.1.](#) 'counter' attribute

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

[2.32.2.](#) 'hangIndent' attribute

Deprecated.

[2.32.3.](#) 'style' attribute

Deprecated.

[2.33.](#) <middle>

Represents the main content of the document.

This element appears as child element of: <rfc> ([Section 2.46](#)).

Content model:

One or more <section> elements ([Section 2.47](#))

[2.34.](#) <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 2.27](#)).

Content model:

One or more <t> elements ([Section 2.53](#))

[2.34.1.](#) 'title' attribute (mandatory)

The title of the note.

**[2.35.](#) **

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 2.17](#)), ([Section 2.31](#)), and <t> ([Section 2.53](#)).

Content model:

One or more elements ([Section 2.31](#))

[2.35.1.](#) '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"

[2.35.2.](#) 'start' attribute

The ordinal value to start the list at. This defaults to "1".

[2.35.3.](#) '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, ...)

[[anchor10: 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.

[2.36.](#) <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.

[2.36.1.](#) 'abbrev' attribute

Abbreviated variant.

[2.36.2.](#) 'ascii' attribute

The ASCII equivalent of the organization's name.

[2.37.](#) <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.

[2.38.](#) <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:

Either:

o In any order:

- * <street> elements ([Section 2.51](#))
- * <city> elements ([Section 2.12](#))
- * <region> elements ([Section 2.45](#))
- * <code> elements ([Section 2.13](#))
- * <country> elements ([Section 2.14](#))

Or:

o One or more <postalLine> elements ([Section 2.39](#))

[2.38.1.](#) 'xml:lang' attribute

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

[2.39.](#) <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 2.38](#)).

Content model: only text content.

2.40. <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:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <spanx> elements ([Section 2.50](#))
- o <tt> elements ([Section 2.57](#))
- o elements ([Section 2.52](#))
- o elements ([Section 2.8](#))
- o elements ([Section 2.21](#))
- o <i> elements ([Section 2.28](#))

2.41. <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:

- o Text

- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <spanx> elements ([Section 2.50](#))
- o <tt> elements ([Section 2.57](#))
- o elements ([Section 2.52](#))
- o elements ([Section 2.8](#))
- o elements ([Section 2.21](#))
- o <i> elements ([Section 2.28](#))

[2.42.](#) <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>
  <innerRefContent>Self-published pamphlet</innerRefContent>
</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 2.43](#)).

Content model:

In any order:

- o Text
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.8](#))
- o `<i>` elements ([Section 2.28](#))
- o `` elements ([Section 2.21](#))
- o `` elements ([Section 2.52](#))

[2.43.](#) `<reference>`

Represents a bibliographical reference.

This element appears as child element of: `<references>` ([Section 2.44](#)).

Content model:

In this order:

1. One `<front>` element ([Section 2.27](#))
2. In any order:
 - * `<seriesInfo>` elements ([Section 2.48](#))
 - * `<format>` elements ([Section 2.26](#))
 - * `<refcontent>` elements ([Section 2.42](#))
 - * `<annotation>` elements ([Section 2.3](#))

[2.43.1.](#) '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.

[2.43.2.](#) '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).

[2.44.](#) <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 2.43](#))

[2.44.1.](#) 'title' attribute

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

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

[2.45.](#) <region>

Provides the region name in a postal address.

This element appears as child element of: <postal> ([Section 2.38](#)).

Content model: only text content.

[2.46.](#) <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 <front> element ([Section 2.27](#))

2. One <middle> element ([Section 2.33](#))
3. One optional <back> element ([Section 2.9](#))

[2.46.1.](#) '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.

[2.46.2.](#) 'category' attribute

Document category (see [Appendix A.1](#)).

Allowed values:

- o "std"
- o "bcp"
- o "info"
- o "exp"
- o "historic"

[2.46.3.](#) 'consensus' attribute

Affects the generated boilerplate.

See [[RFC5741](#)] for more information.

Allowed values:

- o "no"
- o "yes"

[2.46.4.](#) 'docName' attribute

For Internet-Drafts, this specifies the draft name (which appears below the title).

Note that the file extension is not part of the draft, so in general it should end with the current draft number ("-"), plus two digits).

Furthermore, it is good practice to disambiguate current editor

copies from submitted drafts (for instance, by replacing the draft number with the string "latest").

See [[IDGUIDE](#)] for further information.

2.46.5. 'ipr' attribute

Represents the Intellectual Property status of the document. See [Appendix A.2](#) for details.

Allowed values:

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

2.46.6. 'iprExtract' attribute

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

2.46.7. 'number' attribute

The number of the RFC to be produced.

2.46.8. '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 format, hyperlinks can be generated. Also, the value ought to be reformatted to insert whitespace after each comma if not already present.

2.46.9. 'seriesNo' attribute

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

2.46.10. 'sortRefs' attribute

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

Allowed values:

- o "yes"
- o "no" (default)

2.46.11. 'submissionType' attribute

The document stream.

See [Section 2 of \[RFC5741\]](#) for details.

Allowed values:

- o "IETF" (default)
- o "IAB"

- o "IRTF"
- o "independent"

2.46.12. '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:

- o "yes" (default)
- o "no"

2.46.13. 'tocDepth' attribute

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

2.46.14. 'tocInclude' attribute

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

Allowed values:

- o "yes" (default)
- o "no"

2.46.15. '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 format, hyperlinks can be generated. Also, the value ought to be reformatted to insert whitespace after each comma if not already present.

2.46.16. 'xml:lang' attribute

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

See Section 2.12 of [[XML](#)] for more information.

[2.47.](#) <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 2.33](#)), and <section> ([Section 2.47](#)).

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](#))
 - * <figure> elements ([Section 2.25](#))
 - * <texttable> elements ([Section 2.54](#))
 - * <iref> elements ([Section 2.29](#))
 - * <artwork> elements ([Section 2.5](#))
 - * <sourcecode> elements ([Section 2.49](#))
3. Optional <section> elements ([Section 2.47](#))

[2.47.1.](#) 'anchor' attribute

Document-wide unique identifier for this section.

[2.47.2.](#) '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.

[2.47.3.](#) **'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:

- o "yes" (default)
- o "no"

[2.47.4.](#) **'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:

- o "yes"
- o "no" (default)

[2.47.5.](#) **'title' attribute**

Deprecated. Use <titleelement> instead.

[2.47.6.](#) **'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:

- o "include"
- o "exclude"

- o "default" (default)

[2.48.](#) **<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 2.43](#)).

Content model: this element does not have any contents.

[2.48.1.](#) **'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".

[2.48.2.](#) **'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).

[2.49.](#) **<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. To avoid these problems, the artwork needs to be surrounded in a CDATA structure: "<![CDATA[]]". For example:

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

Instead of using a CDATA structure, the authour could instead escape

the "&" and "<" characters as "&" and "<".

This element appears as child element of: <aside> ([Section 2.6](#)), <blockquote> ([Section 2.10](#)), <dd> ([Section 2.17](#)), <figure> ([Section 2.25](#)), ([Section 2.31](#)), <section> ([Section 2.47](#)), and <t> ([Section 2.53](#)).

Content model: only text content.

[2.49.1.](#) '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).

[2.49.2.](#) '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 in the non-canonical output formats. 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:

- o abnf
- o asn.1
- o bash
- o c++
- o c
- o cbor
- o dtd
- o java
- o javascript

- o json
- o mib
- o perl
- o pseudocode
- o python
- o rnc
- o 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.

[2.50.](#) <spanx>

Deprecated. Use , <i>, and <tt> instead.

This element appears as child element of: <annotation> ([Section 2.3](#)), <c> ([Section 2.11](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), and <t> ([Section 2.53](#)).

Content model: only text content.

[2.50.1.](#) 'style' attribute

Deprecated.

[2.50.2.](#) 'xml:space' attribute

Deprecated.

Allowed values:

- o "default"
- o "preserve" (default)

[2.51.](#) <street>

Provides a street address.

This element appears as child element of: <postal> ([Section 2.38](#)).

Content model: only text content.

**2.52. **

Indicates text that is semantically strong. This element will be displayed as bold after processing. This element has the same effects as .

This element appears as child element of: <annotation> ([Section 2.3](#)), <blockquote> ([Section 2.10](#)), <c> ([Section 2.11](#)), <cref> ([Section 2.15](#)), <dd> ([Section 2.17](#)), <dt> ([Section 2.20](#)), ([Section 2.21](#)), <i> ([Section 2.28](#)), ([Section 2.31](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), <refcontent> ([Section 2.42](#)), <t> ([Section 2.53](#)), and <tt> ([Section 2.57](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <tt> elements ([Section 2.57](#))
- o elements ([Section 2.21](#))
- o <i> elements ([Section 2.28](#))

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 2.17](#)), ([Section 2.31](#)), <list> ([Section 2.32](#)), <note> ([Section 2.34](#)), and <section> ([Section 2.47](#)).

Content model:

In any order:

- o Text
- o `<list>` elements ([Section 2.32](#))
- o `` elements ([Section 2.35](#))
- o `` elements ([Section 2.59](#))
- o `<dl>` elements ([Section 2.19](#))
- o `<figure>` elements ([Section 2.25](#))
- o `<artwork>` elements ([Section 2.5](#))
- o `<sourcecode>` elements ([Section 2.49](#))
- o `<xref>` elements ([Section 2.63](#))
- o `<eref>` elements ([Section 2.23](#))
- o `<iref>` elements ([Section 2.29](#))
- o `<cref>` elements ([Section 2.15](#))
- o `<spanx>` elements ([Section 2.50](#))
- o `<vspace>` elements ([Section 2.61](#))
- o `<tt>` elements ([Section 2.57](#))
- o `` elements ([Section 2.52](#))
- o `` elements ([Section 2.8](#))
- o `` elements ([Section 2.21](#))
- o `<i>` elements ([Section 2.28](#))

[2.53.1.](#) **'anchor' attribute**

Document-wide unique identifier for this paragraph.

[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 2.47](#)).

Content model:

In this order:

1. One optional <titleelement> element ([Section 2.56](#))
2. One optional <preamble> element ([Section 2.41](#))
3. One or more <ttcol> elements ([Section 2.58](#))
4. Optional <c> elements ([Section 2.11](#))
5. One optional <postamble> element ([Section 2.40](#))

[2.54.1.](#) 'align' attribute

Determines the horizontal alignment of the table.

Allowed values:

- o "left"
- o "center" (default)
- o "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

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

Allowed values:

- o "all"
- o "none"
- o "headers"
- o "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:

- o "true"
- o "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's long (~40 characters), the "abbrev" attribute is used to specify

an abbreviated variant.

This element appears as child element of: <front> ([Section 2.27](#)).

Content model: only text content.

[2.55.1.](#) 'abbrev' attribute

Specifies an abbreviated variant of the document 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 2.47](#)), and <texttable> ([Section 2.54](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o <tt> elements ([Section 2.57](#))

[2.57.](#) <tt>

Causes the text to be displayed in a constant-width font.

This element appears as child element of: <annotation> ([Section 2.3](#)), ([Section 2.8](#)), <blockquote> ([Section 2.10](#)), <c> ([Section 2.11](#)), <cref> ([Section 2.15](#)), <dd> ([Section 2.17](#)), <dt> ([Section 2.20](#)), ([Section 2.21](#)), <i> ([Section 2.28](#)), ([Section 2.31](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), <refcontent> ([Section 2.42](#)), ([Section 2.52](#)), <t> ([Section 2.53](#)), and <titleelement> ([Section 2.56](#)).

Content model:

In any order:

- o Text
- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o elements ([Section 2.52](#))
- o elements ([Section 2.8](#))
- o elements ([Section 2.21](#))
- o <i> elements ([Section 2.28](#))

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:

- o <xref> elements ([Section 2.63](#))
- o <eref> elements ([Section 2.23](#))
- o <iref> elements ([Section 2.29](#))
- o <cref> elements ([Section 2.15](#))
- o Text

2.58.1. 'align' attribute

Determines the horizontal alignment within the table column.

Allowed values:

- o "left" (default)

- o "center"
- o "right"

[2.58.2.](#) 'width' attribute

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

[2.59.](#)

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

This element appears as child element of: <dd> ([Section 2.17](#)), ([Section 2.31](#)), and <t> ([Section 2.53](#)).

Content model:

One or more elements ([Section 2.31](#))

[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.](#) <vspace>

Deprecated.

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

Content model: this element does not have any contents.

[2.61.1.](#) 'blankLines' attribute

Deprecated.

[2.62.](#) <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 2.46.11](#)).

This element appears as child element of: <front> ([Section 2.27](#)).

Content model: only text content.

[2.63.](#) <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. [[anchor11: 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](#)), ([Section 2.8](#)), <c> ([Section 2.11](#)), <cref> ([Section 2.15](#)), <dd> ([Section 2.17](#)), <dt> ([Section 2.20](#)), ([Section 2.21](#)), <i> ([Section 2.28](#)), ([Section 2.31](#)), <postamble> ([Section 2.40](#)), <preamble> ([Section 2.41](#)), ([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.

2.63.1. 'format' attribute

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

"counter"

Inserts just a counter; this use 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"

This attribute value is deprecated.

"title"

This attribute value is deprecated.

Allowed values:

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

2.63.2. 'pageno' attribute

Deprecated.

Allowed values:

- o "true"
- o "false" (default)

2.63.3. 'relative' attribute

Specifies a relative reference from the main target. [[anchor12: Need more description here about how this will be displayed.]]

2.63.4. '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.

2.63.5. 'sectionFormat' attribute

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

- o "of" (default)

- o "comma"
- o "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.
```

2.63.6. '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. Special Unicode Code Points

Although the current RFC format does not allow non-ASCII Unicode characters ([\[UNICODE\]](#)), some of them can be used to enforce certain behaviors of formatters.

For instance:

non-breaking space (U+00A0)

Represents a space character where no line break should happen. This is frequently used in titles (by excluding certain space characters from the line breaking algorithm, the processor will use the remaining whitespace occurrences for line breaks).

non-breaking hyphen (U+2011)

Similarly, this represents a hyphen character where nevertheless no line breaking ought to occur.

word joiner (U+2060)

Also called "zero width non-breaking space". This can be used to disallow line breaking between two non-whitespace characters.

Note that in order to use these characters by name, they need to be declared either in the Document Type Definition (DTD, [\[XML\]](#), [Section 2.9](#)), or in the "internal subset" ([\[XML\]](#), [Section 2.8](#)), like this:

```
<?xml version="1.0"?>

<!DOCTYPE rfc [

    <!-- declare nbsp and friends -->
    <!ENTITY nbsp      "&#xa0;">
    <!ENTITY nbhy      "&#x2011;">
    <!ENTITY wj        "&#x2060;">
]>
```

4. Internationalization Considerations

This format is based on [\[XML\]](#), thus does not have any issues representing arbitrary Unicode [\[UNICODE\]](#) characters in text content.

However, the current canonical RFC format is restricted to US-ASCII [\[USASCII\]](#) characters ([\[RFC2223\]](#), [Section 3](#)). Future versions are likely to relax this role, and it is expected that the vocabulary will be extended so that US-ASCII alternatives can be provided when that makes sense (for instance, in contact information).

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 probalby 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.

Furthermore, 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 either transform xml2rfc to output formats 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

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)

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" formatter.

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] Freed, N. and J. Reschke, "The 'XML2RFC' version 2 Vocabulary", [draft-reschke-xml2rfc](#) (work in progress), January 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>>.

- [RFC2026] Bradner, S., "The Internet Standards Process -- Revision 3", [BCP 9](#), [RFC 2026](#), October 1996.
- [RFC2223] Postel, J. and J. Reynolds, "Instructions to RFC Authors", [RFC 2223](#), October 1997.
- [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>>.
- [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/>>.
- [USASCII] American National Standards Institute, "Coded Character Set -- 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.
- [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>](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>](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

As described in [\[RFC6949\]](#), the v3 format will be the format for canonical RFCs. The format will also likely be allowed for submission of Internet Drafts. The format for both of these tasks is the same; however, software processing an XML file for creation of an

Internet Draft will act differently than it will for creation of what is expected to be an RFC.

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.

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 most work from the RNG, not from a DTD. In fact, the author believes that some of the features of the v3 grammar cannot be specified as a DTD.

This section differentiates between three types of processors:

- o RFC Processor -- The processor used by the RFC Editor to validate the contents of the intended RFC and to create the non-canonical formats that will be published by the RFC Editor.
- o Draft Processor -- The processor used by the IETF to validate the contents of XML that is submitted and to create the other formats that will be supported for Internet Drafts; to produce interim versions of the non-canonical formats so that authors can see how their XML might later be rendered; and to convert documents that use the v2 format to v3.
- o 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 formats so that authors can see how their XML might later be rendered; to create documents in formats different than those supported by the RFC Editor; 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. 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. 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.

Users of current and older XML processors for Internet Drafts and RFCs may be used to them having unhelpful error messages. Beginning with v3, the RFC Processor and Draft Processor will have much better

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.

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 format, 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 is done from `xml.resource.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://xml.resource.org/public/rfc/
    bibxml/reference.RFC.2119.xml"/>
  <xi:include href="http://xml.resource.org/public/rfc/
    bibxml/reference.RFC.4869.xml"/>
  <xi:include href="http://xml.resource.org/public/rfc/
    bibxml/reference.RFC.7169.xml"/>
</references>
```

`<xi:include>` can be used anywhere where 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"/>
```

[[anchor18: Need to think a bit more about using `xi:include` to replace `src=` for artwork or sourcecode.]]

B.2. The RFC Processor

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 (PIs)
- o an external DTD
- o ENTITYs that have a SYSTEM component
- o <artwork> elements with a "src" attribute that points to a URI with a scheme other than "data:"
- o <cref> elements

The RFC Processor will also have a mode that gives hard errors on all of the above other than <cref> elements. This 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 "autogeneratedBoilerplateText", "autogeneratedSectionNumber", "autogeneratedFigureNumber", and "autogeneratedTableNumber" attributes in <rfc>, <section>, <figure>, and <texttable>, respectively.

B.3. The Draft Processor

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 restrictive mode, but that instead gives warnings instead of hard errors.

B.4. 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:

- o 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:

- o comments -- When set to "yes", causes all <cref> elements to be rendered in the output files. The default is "yes".
- o 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 C. 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 }?,
  front,
  middle,
  back?
}
front =
  element front {
    title, author+, date?, area*, workgroup*, keyword*, abstract?,
note*
  }
title =
  element title {
    attribute abbrev { 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 { text }
city = element city { text }
region = element region { text }
code = element code { text }
country = element country { text }
postalLine = element postalLine { 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 | figure | texttable | iref | artwork | sourcecode)*,
    section*
  }
titleelement =
  element titleelement { (text | xref | eref | iref | cref | tt)* }
t =
  element t {
    attribute anchor { xsd:ID }?,
    (text
    | \list
    | ol
    | ul
    | dl
    | figure
    | artwork
    | sourcecode
    | xref
    | eref
    | iref
    | cref
    | spanx
    | vspace
    | tt
    | strong
    | b
    | em
    | i)*
  }
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)*
}
\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 }?,
    [ 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)+
  }
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)*
  }
dd =
  element dd {
    t+
    | (text,
      ol,
      ul,
      dl,
      figure,
      artwork,
      sourcecode,
      xref,
      eref,
      iref,
      cref,
      tt,
      strong,
      b,
      em,
      i)+
  }
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)*
    }
  tt =
    element tt {
      (text | xref | eref | iref | cref | strong | b | em | i)*
    }
  b = element b { (text | xref | eref | iref | cref | tt | em | i)* }
  strong =
    element strong { (text | xref | eref | iref | cref | tt | em | i)*
  }
  i = element i { (text | xref | eref | iref | cref | tt | strong |
  b)* }
  em =
    element em { (text | xref | eref | iref | cref | tt | strong | b)*
  }
  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)*
  }
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)*
}
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)*
  }
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 },
      empty
    }
  format =
    element format {
      attribute target { text }?,
      attribute type { text },
      attribute octets { text }?,
      empty
    }
  annotation =
    element annotation {
      (text
      | xref
      | eref
      | iref
      | cref
      | spanx
      | tt
      | strong
      | b
      | em
      | i)*
    }
  refcontent = element refcontent { (text | tt | b | i | em | strong)*
}
start = rfc
```


[Appendix D](#). 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 }?,
+   front,
+   middle,
+   back?
+ }
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 }?,
    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 { street+, (city | region | code |
- country)* }
+ 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 { text }
city = element city { text }
region = element region { text }
code = element code { text }
country = element country { text }
+ postallLine = element postallLine { 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 | 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 }?,
            (text
+           | \list
+           | ol
+           | ul
+           | dl
            | figure
+           | artwork
+           | sourcecode
            | xref
            | eref
            | iref
            | cref
            | spanx
-           | vspace)*
+           | vspace
+           | tt
+           | strong
+           | b
+           | em
+           | i)*
+       }
+ 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)*
+   }
    \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 }?,
+     [ 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)+
+   }
+ 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)*

```



```

+   }
+ dd =
+   element dd {
+     t+
+     | (text,
+       ol,
+       ul,
+       dl,
+       figure,
+       artwork,
+       sourcecode,
+       xref,
+       eref,
+       iref,
+       cref,
+       tt,
+       strong,
+       b,
+       em,
+       i)+
+   }
+ 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)*
    }
+ tt =
+   element tt {
+       (text | xref | eref | iref | cref | strong | b | em | i)*
+   }
+ b = element b { (text | xref | eref | iref | cref | tt | em | i)*
+ }
+ strong =
+   element strong { (text | xref | eref | iref | cref | tt | em |
+ i)* }
+ i = element i { (text | xref | eref | iref | cref | tt | strong |
+ b)* }
+ em =
+   element em { (text | xref | eref | iref | cref | tt | strong |
+ b)* }
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)*
+    }
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)*
+    }
+  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)*
+    }
+  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 },
      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 {
+     (text
+     | xref
+     | eref
+     | iref
+     | cref
+     | spanx
+     | tt
+     | strong
+     | b
```



```
+      | em
+      | i)*
+    }
+ refcontent = element refcontent { (text | tt | b | i | em |
+ strong)* }
+ start = rfc
```

Index

A

- abbrev attribute
 - in organization element 33
 - in title element 52
- abstract element 9
 - inside front 27
- address element 9
 - inside author 14
- align attribute
 - in artwork element 12
 - in figure element 25
 - in texttable element 50
 - in ttable element 53
- alt attribute
 - in artwork element 12
 - in figure element 26
- anchor attribute
 - in blockquote element 17
 - in cref element 19
 - in figure element 26
 - in reference element 37
 - in section element 43
 - in t element 49
 - in texttable element 50
- annotation element 10
 - inside reference 37
- application/rfc+xml Media Type 60
- area element 11
 - inside front 27
- artwork element 11
 - align attribute 12
 - alt attribute 12
 - height attribute 12
 - inside aside 14
 - inside blockquote 16
 - inside dd 20
 - inside figure 25
 - inside li 30

- inside section 43
- inside t 49
- name attribute 12
- src attribute 12
- type attribute 13
- width attribute 13
- xml:lang attribute 13
- xml:space attribute 13
- ascii attribute
 - in author element 15
 - in email element 24
 - in organization element 33
- aside element 13
 - inside section 43
- Attributes
 - abbrev 33, 52
 - align 12, 25, 50, 53
 - alt 12, 26
 - anchor 17, 19, 26, 37, 43, 49-50
 - ascii 15, 24, 33
 - autogeneratedBoilerplateText 39
 - autogeneratedFigureNumber 26
 - autogeneratedSectionNumber 43
 - autogeneratedTableNumber 50
 - blankLines 55
 - category 39
 - cite 17
 - consensus 39
 - counter 31
 - day 20
 - docName 39
 - empty 54
 - format 56
 - from 21
 - fullname 15
 - hangIndent 31
 - hanging 22
 - height 12, 26
 - initials 15
 - ipr 40
 - iprExtract 41
 - item 29
 - month 20
 - name 12, 45-46
 - number 41
 - numbered 44
 - obsoletes 41
 - octets 27

- pageno 57
- primary 29
- relative 57
- removeInRFC 44
- role 15
- section 57
- sectionFormat 57
- seriesNo 41
- sortRefs 41
- source 19
- spacing 22, 32, 54
- src 12, 26
- start 32
- style 31-32, 47, 51
- subitem 29
- submissionType 41
- suppress-title 26, 51
- surname 15
- symRefs 42
- target 24, 27, 37, 58
- title 26, 31, 38, 44, 51
- to 21
- toc 44
- tocDepth 42
- tocInclude 42
- type 13, 27, 46
- updates 42
- value 45
- width 13, 26, 54
- xml:lang 13, 15, 34, 42
- xml:space 13, 47
- year 20
- author element 14
 - ascii attribute 15
 - fullname attribute 15
 - initials attribute 15
 - inside front 27
 - role attribute 15
 - surname attribute 15
 - xml:lang attribute 15
- autogeneratedBoilerplateText attribute
 - in rfc element 39
- autogeneratedFigureNumber attribute
 - in figure element 26
- autogeneratedSectionNumber attribute
 - in section element 43
- autogeneratedTableNumber attribute
 - in texttable element 50

B

- b element 15
 - inside annotation 10
 - inside blockquote 17
 - inside c 18
 - inside cref 19
 - inside dd 21
 - inside dt 23
 - inside em 24
 - inside i 28
 - inside li 30
 - inside postamble 35
 - inside preamble 36
 - inside refcontent 37
 - inside t 49
 - inside tt 53
- back element 16
 - inside rfc 39
- blankLines attribute
 - in vspace element 55
- blockquote element 16
 - anchor attribute 17
 - cite attribute 17

C

- c element 17
 - inside texttable 50
- category attribute
 - in rfc element 39
- cite attribute
 - in blockquote element 17
- city element 18
 - inside postal 34
- code element 18
 - inside postal 34
- consensus attribute
 - in rfc element 39
- counter attribute
 - in list element 31
- country element 18
 - inside postal 34
- cref element 18
 - anchor attribute 19
 - inside annotation 10
 - inside b 16
 - inside c 17
 - inside dd 21
 - inside dt 23

- inside em 23
- inside i 28
- inside li 30
- inside postamble 35
- inside preamble 36
- inside strong 48
- inside t 49
- inside titleelement 52
- inside tt 53
- inside ttcol 53
- source attribute 19

D

- date element 19
 - day attribute 20
 - inside front 27
 - month attribute 20
 - year attribute 20
- day attribute
 - in date element 20
- dd element 20
 - inside dl 22
- displayreference element 21
 - from attribute 21
 - inside back 16
 - to attribute 21
- dl element 21
 - hanging attribute 22
 - inside dd 20
 - inside li 30
 - inside t 49
 - spacing attribute 22
- docName attribute
 - in rfc element 39
- dt element 22
 - inside dl 22

E

Elements

- abstract 9, 27
- address 9, 14
- annotation 10, 37
- area 11, 27
- artwork 11, 14, 16, 20, 25, 30, 43, 49
- aside 13, 43
- author 14, 27
- b 10, 15, 17-19, 21, 23-24, 28, 30, 35-37, 49, 53
- back 16, 39

blockquote 16
c 17, 50
city 18, 34
code 18, 34
country 18, 34
cref 10, 16-18, 21, 23, 28, 30, 35-36, 48-49, 52-53
date 19, 27
dd 20, 22
displayreference 16, 21
dl 20-21, 30, 49
dt 22
em 10, 16-19, 21, 23, 30, 35-37, 48-49, 53
email 10, 24
eref 10, 16-17, 19-20, 23-24, 28, 30, 35-36, 48-49, 52-53
facsimile 9, 24
figure 14, 16, 20, 25, 30, 43, 49
format 27, 37
front 27, 37-38
i 10, 16-19, 21, 23, 28, 30, 35-37, 48-49, 53
iref 10, 14, 16-17, 21, 23, 25, 28, 30, 35-36, 43, 48-49, 52-53
keyword 27, 29
li 29, 32, 54
list 30, 49
middle 31, 39
note 28, 31
ol 20, 30, 32, 49
organization 14, 33
phone 9, 33
postal 9, 34
postalLine 34
postamble 25, 35, 50
preamble 25, 35, 50
refcontent 36-37
reference 37-38
references 16, 38
region 34, 38
rfc 38
section 16, 31, 43
seriesInfo 37, 45
sourcecode 14, 17, 20, 25, 30, 43, 45, 49
spanx 10, 17, 35-36, 47, 49
street 34, 47
strong 10, 17, 19, 21, 23, 28, 30, 35-37, 48-49, 53
t 9, 14, 20, 30-31, 43, 48
texttable 14, 43, 49
title 27, 51
titleelement 25, 43, 50, 52

- tt 10, 16-17, 19, 21, 23, 28, 30, 35-37, 48-49, 52
- ttcol 50, 53
- ul 20, 30, 49, 54
- uri 10, 54
- vspace 49, 55
- workgroup 27, 55
- xref 10, 16-17, 19-20, 22-23, 28, 30, 35-36, 48-49, 52-53, 55
- em element 23
 - inside annotation 10
 - inside b 16
 - inside blockquote 17
 - inside c 18
 - inside cref 19
 - inside dd 21
 - inside dt 23
 - inside li 30
 - inside postamble 35
 - inside preamble 36
 - inside refcontent 37
 - inside strong 48
 - inside t 49
 - inside tt 53
- email element 24
 - ascii attribute 24
 - inside address 10
- empty attribute
 - in ul element 54
- eref element 24
 - inside annotation 10
 - inside b 16
 - inside c 17
 - inside cref 19
 - inside dd 20
 - inside dt 23
 - inside em 23
 - inside i 28
 - inside li 30
 - inside postamble 35
 - inside preamble 36
 - inside strong 48
 - inside t 49
 - inside titleelement 52
 - inside tt 53
 - inside ttcol 53
 - target attribute 24

F

facsimile element 24

- inside address 9
- figure element 25
 - align attribute 25
 - alt attribute 26
 - anchor attribute 26
 - autogeneratedFigureNumber attribute 26
 - height attribute 26
 - inside aside 14
 - inside blockquote 16
 - inside dd 20
 - inside li 30
 - inside section 43
 - inside t 49
 - src attribute 26
 - suppress-title attribute 26
 - title attribute 26
 - width attribute 26
- format attribute
 - in xref element 56
- format element 27
 - inside reference 37
 - octets attribute 27
 - target attribute 27
 - type attribute 27
- from attribute
 - in displayreference element 21
- front element 27
 - inside reference 37
 - inside rfc 38
- fullname attribute
 - in author element 15

H

- hangIndent attribute
 - in list element 31
- hanging attribute
 - in dl element 22
- height attribute
 - in artwork element 12
 - in figure element 26

I

- i element 28
 - inside annotation 10
 - inside b 16
 - inside blockquote 17
 - inside c 18
 - inside cref 19

- inside dd 21
- inside dt 23
- inside li 30
- inside postamble 35
- inside preamble 36
- inside refcontent 37
- inside strong 48
- inside t 49
- inside tt 53
- initials attribute
 - in author element 15
- ipr attribute
 - '*2026' 66
 - '*3667' 66
 - '*3978' 66
 - '*trust200811' 66
 - '*trust200902' 64
 - 'noDerivativesTrust200902' 65
 - 'noModificationTrust200902' 65
 - 'pre5378Trust200902' 65
 - 'trust200902' 65
 - in rfc element 40
- iprExtract attribute
 - in rfc element 41
- iref element 28
 - inside annotation 10
 - inside aside 14
 - inside b 16
 - inside c 17
 - inside dd 21
 - inside dt 23
 - inside em 23
 - inside figure 25
 - inside i 28
 - inside li 30
 - inside postamble 35
 - inside preamble 36
 - inside section 43
 - inside strong 48
 - inside t 49
 - inside titleelement 52
 - inside tt 53
 - inside ttcoll 53
 - item attribute 29
 - primary attribute 29
 - subitem attribute 29
- item attribute
 - in iref element 29

K

- keyword element 29
 - inside front 27

L

- li element 29
 - inside ol 32
 - inside ul 54
- list element 30
 - counter attribute 31
 - hangIndent attribute 31
 - inside t 49
 - style attribute 31

M

- Media Type
 - application/rfc+xml 60
- middle element 31
 - inside rfc 39
- month attribute
 - in date element 20

N

- name attribute
 - in artwork element 12
 - in seriesInfo element 45
 - in sourcecode element 46
- note element 31
 - inside front 28
 - title attribute 31
- number attribute
 - in rfc element 41
- numbered attribute
 - in section element 44

O

- obsoletes attribute
 - in rfc element 41
- octets attribute
 - in format element 27
- ol element 32
 - inside dd 20
 - inside li 30
 - inside t 49
 - spacing attribute 32
 - start attribute 32
 - style attribute 32
- organization element 33

- abbrev attribute 33
- ascii attribute 33
- inside author 14

P

- pageno attribute
 - in xref element 57
- phone element 33
 - inside address 9
- postal element 34
 - inside address 9
 - xml:lang attribute 34
- postalline element 34
 - inside postal 34
- postamble element 35
 - inside figure 25
 - inside texttable 50
- preamble element 35
 - inside figure 25
 - inside texttable 50
- primary attribute
 - in iref element 29

R

- refcontent element 36
 - inside reference 37
- reference element 37
 - anchor attribute 37
 - inside references 38
 - target attribute 37
- references element 38
 - inside back 16
 - title attribute 38
- region element 38
 - inside postal 34
- relative attribute
 - in xref element 57
- removeInRFC attribute
 - in section element 44
- rfc element 38
 - autogeneratedBoilerplateText attribute 39
 - category attribute 39
 - consensus attribute 39
 - docName attribute 39
 - ipr attribute 40
 - iprExtract attribute 41
 - number attribute 41
 - obsoletes attribute 41

- seriesNo attribute 41
- sortRefs attribute 41
- submissionType attribute 41
- symRefs attribute 42
- tocDepth attribute 42
- tocInclude attribute 42
- updates attribute 42
- xml:lang attribute 42
- role attribute
 - in author element 15

S

- section attribute
 - in xref element 57
- section element 43
 - anchor attribute 43
 - autogeneratedSectionNumber attribute 43
 - inside back 16
 - inside middle 31
 - inside section 43
 - numbered attribute 44
 - removeInRFC attribute 44
 - title attribute 44
 - toc attribute 44
- sectionFormat attribute
 - in xref element 57
- seriesInfo element 45
 - inside reference 37
 - name attribute 45
 - value attribute 45
- seriesNo attribute
 - in rfc element 41
- sortRefs attribute
 - in rfc element 41
- source attribute
 - in cref element 19
- sourcecode element 45
 - inside aside 14
 - inside blockquote 17
 - inside dd 20
 - inside figure 25
 - inside li 30
 - inside section 43
 - inside t 49
 - name attribute 46
 - type attribute 46
- spacing attribute
 - in dl element 22

- in ol element 32
 - in ul element 54
- spanx element 47
 - inside annotation 10
 - inside c 17
 - inside postamble 35
 - inside preamble 36
 - inside t 49
 - style attribute 47
 - xml:space attribute 47
- src attribute
 - in artwork element 12
 - in figure element 26
- start attribute
 - in ol element 32
- street element 47
 - inside postal 34
- strong element 48
 - inside annotation 10
 - inside blockquote 17
 - inside c 17
 - inside cref 19
 - inside dd 21
 - inside dt 23
 - inside em 23
 - inside i 28
 - inside li 30
 - inside postamble 35
 - inside preamble 36
 - inside refcontent 37
 - inside t 49
 - inside tt 53
- style attribute
 - in list element 31
 - in ol element 32
 - in spanx element 47
 - in texttable element 51
- subitem attribute
 - in iref element 29
- submissionType attribute
 - in rfc element 41
- suppress-title attribute
 - in figure element 26
 - in texttable element 51
- surname attribute
 - in author element 15
- symRefs attribute
 - in rfc element 42

T

- t element 48
 - anchor attribute 49
 - inside abstract 9
 - inside aside 14
 - inside dd 20
 - inside li 30
 - inside list 31
 - inside note 31
 - inside section 43
- target attribute
 - in eref element 24
 - in format element 27
 - in reference element 37
 - in xref element 58
- texttable element 49
 - align attribute 50
 - anchor attribute 50
 - autogeneratedTableNumber attribute 50
 - inside aside 14
 - inside section 43
 - style attribute 51
 - suppress-title attribute 51
 - title attribute 51
- title attribute
 - in figure element 26
 - in note element 31
 - in references element 38
 - in section element 44
 - in texttable element 51
- title element 51
 - abbrev attribute 52
 - inside front 27
- titleelement element 52
 - inside figure 25
 - inside section 43
 - inside texttable 50
- to attribute
 - in displayreference element 21
- toc attribute
 - in section element 44
- tocDepth attribute
 - in rfc element 42
- tocInclude attribute
 - in rfc element 42
- tt element 52
 - inside annotation 10
 - inside b 16

- inside blockquote 17
- inside c 17
- inside cref 19
- inside dd 21
- inside dt 23
- inside em 23
- inside i 28
- inside li 30
- inside postamble 35
- inside preamble 36
- inside refcontent 37
- inside strong 48
- inside t 49
- inside titleelement 52
- ttcol element 53
 - align attribute 53
 - inside texttable 50
 - width attribute 54
- type attribute
 - in artwork element 13
 - in format element 27
 - in sourcecode element 46

U

- ul element 54
 - empty attribute 54
 - inside dd 20
 - inside li 30
 - inside t 49
 - spacing attribute 54
- updates attribute
 - in rfc element 42
- uri element 54
 - inside address 10

V

- value attribute
 - in seriesInfo element 45
- vspace element 55
 - blankLines attribute 55
 - inside t 49

W

- width attribute
 - in artwork element 13
 - in figure element 26
 - in ttcol element 54
- workgroup element 55

inside front 27

X

- xml:lang attribute
 - in artwork element 13
 - in author element 15
 - in postal element 34
 - in rfc element 42
- xml:space attribute
 - in artwork element 13
 - in spanx element 47
- xref element 55
 - format attribute 56
 - inside annotation 10
 - inside b 16
 - inside c 17
 - inside cref 19
 - inside dd 20
 - inside dt 22
 - inside em 23
 - inside i 28
 - inside li 30
 - inside postamble 35
 - inside preamble 36
 - inside strong 48
 - inside t 49
 - inside titleelement 52
 - inside tt 53
 - inside ttcol 53
 - pageno attribute 57
 - relative attribute 57
 - section attribute 57
 - sectionFormat attribute 57
 - target attribute 58
- xref formats
 - counter 56
 - default 56
 - none 56
 - title 57

Y

- year attribute
 - in date element 20

Author's Address

Paul Hoffman
VPN Consortium

EMail: paul.hoffman@vpnc.org