SVG Drawings for RFCs: SVG 1.2 RFC
draft-brownlee-svg-rfc-13

Abstract

This document specifies SVG 1.2 RFC - an SVG profile for use in
diagrams that may appear in RFCs - and considers some of the issues
concerning the creation and use of such diagrams.

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 April 18, 2016.

Copyright Notice

Copyright (c) 2015 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.
1. Introduction

Over the last two years the RFC Editor has worked with the Internet community to develop specifications for changes in the format of RFCs. An outline of the resulting specifications was published as [RFC6949] in May 2013. Since then a Design Team has been working with the RFC Editor to flesh out those specifications. One aspect of the changes is to allow line drawings in RFCs; [RFC6949] says

"Graphics may include ASCII art and a more complex form to be defined, such as SVG line art [SVG]. Color and grayscale will not be accepted. RFCs must correctly display in monochromatic black-and-white to allow for monochrome displays, black-and-white printing, and support for visual disabilities."

SVG (Scalable Vector Graphics) has been developed by W3C, the World Wide Web Consortium; its current standard is SVG 1.1 Full [W3C.REC-SVG11-20110816]. This document defines SVG 1.2 RFC, an SVG profile (i.e. a subset of SVG) that is suitable for RFC line drawings.

Note that in RFCs, the text provides normative descriptions of protocols, systems, etc. Diagrams may be used to help explain concepts more clearly, but they provide supporting detail, and should not be considered to be complete specifications in themselves.
The details (particularly any vocabularies) described in this document are expected to change based on experience gained in implementing the RFC production center’s toolset. Revised documents will be published capturing those changes as the toolset is completed. Other implementors must not expect those changes to remain backwards-compatible with the details described in this document.

2. SVG 1.2 RFC: An SVG profile for RFCs

As a starting point for SVG 1.2 RFC, the Design Team decided to use SVG 1.2 Tiny [W3C.REC-SVGTiny12-20081222]. SVG 1.2 Tiny is an SVG subset intended to be implemented on small, mobile devices such as cellphones and smartphones. That should allow RFCs to be rendered well on such devices, especially those that have small screens. However, RFCs are self-contained documents that do not change once they are published. The use of SVG drawings in RFCs is intended to allow authors to create drawings that are simple to produce, and easier to understand than our traditional ‘ASCII Art’ ones. In short, we are also trying to improve access to the content in RFCs, so SVG drawings need to be kept as simple as possible.

SVG can provide a complete User Interface, but within RFCs, all we need are simple diagrams that do not change once the RFC is published. Therefore, SVG RFC does not allow anything from the following sections in SVG Tiny 1.2 [W3C.REC-SVGTiny12-20081222]:

12  Multimedia
13  Interactivity
15  Scripting
16  Animation
18  Metadata
19  Extensibility

Note that SVG Tiny 1.2 elements may have many properties or attributes that are needed to support aspects of the above sections. Those are not allowed in SVG 1.2 RFC.

Considering the other sections in SVG Tiny 1.2 [W3C.REC-SVGTiny12-20081222]:

9  Basic Shapes
10  Text
    Everything in these sections is allowed in SVG 1.2 RFC.

11  Painting: Filling, Stroking, Colors and Paint Servers
14 Linking
SVG Tiny 1.2 allows internationalized IRIs in references. In
SVG 1.2 RFC such links must be ASCII only. That should not
cause problems, since one can just use the URI form of any IRI.
Authors should try to use links only to URIs that are long-term
stable.

17 Fonts
SVG 1.2 RFC only allows ‘serif’, ‘sans-serif’ and ‘monospace’
generic font families from the WebFonts facility, described in
CSS 2.1, [W3C.REC-CSS2-20110607], section 15, Fonts. In
particular, the SVG ‘font’ element is not allowed.

2.1. Elements, properties and attributes allowed in SVG 1.2 RFC

Elements, properties and attributes selected for SVG 1.2 RFC from
[W3C.REC-SVGTiny12-20081222].

In the list below, elements and properties are listed on the
left, and their allowed values are given in parentheses on the
right.

<color> is the list of allowed colors, a black-and-white
subset of the SVG color names.

<style> is a set of CSS attributes that are commonly used (by SVG
drawing applications). They are not part of SVG Tiny 1.2,
but are included here for compatibility. Note that
- There is no guarantee that any renderer will implement
  all the CSS attributes a drawing application may use.
- Authors will need to consider the compatibility of their
drawings with rendering devices.

Elements:

<color> is the list of allowed colors, a black-and-white
subset of the SVG color names.

<style> is a set of CSS attributes that are commonly used (by SVG
drawing applications). They are not part of SVG Tiny 1.2,
but are included here for compatibility. Note that
- There is no guarantee that any renderer will implement
  all the CSS attributes a drawing application may use.
- Authors will need to consider the compatibility of their
drawings with rendering devices.

Elements:

<color> is the list of allowed colors, a black-and-white
subset of the SVG color names.

<style> is a set of CSS attributes that are commonly used (by SVG
drawing applications). They are not part of SVG Tiny 1.2,
but are included here for compatibility. Note that
- There is no guarantee that any renderer will implement
  all the CSS attributes a drawing application may use.
- Authors will need to consider the compatibility of their
drawings with rendering devices.

Elements:
use (x, y, href, xlink:href, id, role, fill, transform)
rect (x, y, width, height, rx, ry, stroke-miterlimit id, role, fill, <style>, transform)
circle (cx, cy, r, id, role, fill, <style>, transform)
ellipse (cx, cy, rx, ry, id, role, fill, <style>, transform)
line (x1, y1, x2, y2, id, role, fill, transform)
polyline (points, id, role, fill, transform)
polygon (points, id, role, fill, <style>, transform)
text (x, y, rotate, id, role, fill, <style>, transform)
tspan (id, role, fill)
textArea (x, y, width, height, auto, id, role, fill, transform)
tbreak (id, role)
solidColor (id, role, fill)
linearGradient (gradientUnits, x1, y1, x2, y2, id, role, fill)
radialGradient (gradientUnits, cx, cy, r, id, role, fill)
stop (id, role, fill)
path (d, pathLength, stroke-miterlimit, id, role, fill, <style>, transform)

Properties: (most allow inherit as a value)

<style> (font-family, font-weight, font-style, font-variant, direction, unicode-bidi, text-anchor, fill, fill-rule)
<color> (black, white, #000000, #ffffff, #FFFFFF)
stroke (<color>, none, currentColor)
stroke-width
stroke-linecap (butt, round, square)
stroke-linejoin (miter, round, bevel)
stroke-miterlimit
stroke-dasharray
stroke-dashoffset
stroke-opacity
vector-effect (non-scaling-stroke, none)
viewport-fill (none, currentColor)
viewport-fill-opacity
display (inline, block, list-item, run-in, compact, marker, table, inline-table, table-row-group, table-header-group, table-footer-group, table-row, table-column-group, table-column, table-cell, table-caption,
3. How to create SVG drawings

Many drawing packages can be used to create SVG drawings, for example Open Source packages Inkscape and Dia. Be aware that such packages may use SVG elements or attributes that are not allowed in SVG 1.2 RFC.

- For example, the ‘marker’ attribute is often used to place symbols such as arrowheads on lines, but ‘marker’ is not allowed in SVG 1.2 Tiny or SVG 1.2 RFC. In such cases one has to draw the arrowhead in another, simpler way.

- SVG clip paths are used to define a shape; objects outside that shape become invisible. The ‘clipPath’ element is not allowed in SVG 1.2 Tiny or SVG 1.2 RFC.
Diagrams produced with these packages may contain elements, their attributes or properties, or values of attributes or properties that are not allowed in SVG 1.2 RFC. We will need to provide a tool to strip out anything that is not allowed in SVG 1.2 RFC, or to replace disallowed values, e.g., 'sans-serif' for 'Sans' as values for 'font-family'. Experience with a simple test version of a tool for this has shown that such deletion and replacement can be effective for making SVG files from drawing packages conform to SVG 1.2 RFC, without visibly changing the diagrams they produce.

The tool described above can also be used by Authors simply to check that their diagrams conform to SVG 1.2 RFC. To help with this, if visible changes do occur, the tool should produce a list of non-allowed keywords and the context in which they were found.

Another way to create SVG drawings is to write programs to draw them. For example, using python and its svgwrite module is a pleasant environment (for those who like writing code).

To include a diagram into an RFC, the xml2rfc (v3) tool will need to provide a way to include SVG drawings in Internet Drafts, as described in [XML2RFCv3] sections 2.5 and 2.67.

4. Accessibility Considerations

One of the long-term goals for RFCs is to make them more accessible, e.g. to sight-impaired readers. For diagrams, it would be useful for authors to provide alternative forms of the diagram, so that voice-reading software could be used to 'talk through' the diagram. Simply reading the SVG code for a complex diagram seems unlikely to work.

SVG 1.2 RFC allows SVG’s ‘title’ and ‘desc’ elements. ‘title’ provides a brief text caption for an SVG object (much like a figure caption), and ‘desc’ provides a longer text description of what the object actually represents. As well, the SVG ‘role’ attribute can be used to indicate to a browser how an SVG object is to be interpreted. Good suggestions on how to use these elements are given in [SVG-ACCESS-TIPS].

ARIA is a W3C Recommendation for using SVG to create ‘Accessible Rich Internet Applications.’ A helpful introduction to ARIA is provided by [SVG-ARIA-PRIMER], while [SVG-USING-ARIA] gives examples of how to use ARIA to enhance SVG accessibility.
5. Meta-language for diagrams common in RFCs

This section presents a few examples of possible meta-languages which could be used to create the kinds of diagrams that are most common in RFCs. Note that they are merely examples, they do not imply that these particular experimental languages might be more widely implemented or used. Instead, they seem to show that designing meta-languages simple enough to serve as audible representations of complex diagrams is difficult indeed!

The SVG diagrams produced from the following examples can be seen at [This-I-D-with-SVG-included] along with an html version of this draft that includes the SVG diagrams.

5.1. Packet Layout Diagrams

Example: Figure 3 from [RFC0793].

In these examples the first line specifies the generated SVG filename. The scale factor determines the size of the SVG drawing; it needs to be set so that the drawing fits nicely into the final document.

‘packet;’ starts the packet description; it’s followed by a description of the fields in each row.
5.2. Sequence Diagrams (1)

Example: Figure 6 from [ExpTrustedProxy].

In this example, columns are vertical lines with a text header above them. There are three columns, and columns 1 and 2 are spaced 250 pixels apart.

The rest of the file describes objects to be drawn; most of them are polylines (polygons) from one column to another, but object 3 only extends across to 0.3 of the distance between columns 1 and 2.
column 1 width 250;  # columns have vertical line to bottom
text above "user-agent";

column 2 width 250;
text "Proxy";

column 3;  # Last col
text "Server";

object 1;  # Only need polylines
pline 1 to 2, arrowhead at end;
text above "(1) TLS ClientHello";
text below "(ALPN ProtocolName: http)";

object 2;
pline 1 to 2, arrowhead at start;
text above "(2) TLS Error";
text below "(Proxy Cert)";

object 3;
pline 1 to 1.3, down, back to 1, arrowhead at end;
text seg 2 centre "(inform user of the SecureProxy)";

object 4;
pline 1 to 2, arrowhead at end;
text above "(3) TLS ClientHello";

object 5;
pline 1 to 2, arrowhead at start;
text above "(4) ServerHello";

object 6;
blank 1 to 2;

object 7;
block 1 to 2, objects 8 to 15;
text above "HTTP2.0";

object 8;
pline 1 to 2, arrowhead at end;
text seg 1 centre "(5) stream(X) GET";

object 9;
pline 2 to 3, arrowhead at end;
text seg 1 above "(6) TLS ClientHello";

object 10;
pline 2 to 3, arrowhead at start;
text seg 1 above "TLS ServerHello";

object 11;
  blank 2 to 3;

object 12;
  block 2 to 3, objects 13 to 15;
  text seg 1 above "HTTP2.0";

object 13;
  pline 2 to 3, arrowhead at end;
  text seg 1 centre "(7) stream(Z) GET";

object 14;
  pline 2 to 3, arrowhead at start;
  text seg 1 centre "(8) stream(Z) 200 OK";

object 15;
  pline 1 to 2, arrowhead at start;
  text seg 1 centre "(9) stream(X) 200 OK";

5.3. Sequence Diagrams (2)

Example: Figure 3 from [RFC4321].

This example uses (x,y) coordinates to specify points in plines. For these, the x units are columns and the y units are lines (positive means ‘down the diagram’).

both x and y may be absolute, e.g. 4.3, or relative, e.g. +1.5). For the first point of a pline, relative means ‘relative to the starting point of the previous pline,’ for other points in a pline it means ‘relative to the last point.’

Note that column 1 is drawn in white, i.e. nothing is drawn for it. It’s simply used to make a blank area where objects 8 and 9 can place text. For both those objects a pline is used to specify the text’s position.

Last, the metalanguage allows simple macros, introduced by ‘define foo = ’. These make it easier to re-use definitions, for example of line types.

info;
  output "rfc4321-fig3.svg", scale 0.9;

# Sat, 5 Apr 14 (NZDT)
#Figure 3 of RFC 4321

define hw = width 110;  # Hop width

column 1 width 130, color "white";  # No heading or vertical line

column 2 hw;  text above "UAC";

column 3 hw;  text "P1";

column 4 hw;  text "P2";

column 5 hw;  text "P3";

column 6 hw;  text "UAS";

define tgrey = width 5;  # Thick grey
define ahe = arrowhead at end;

object 1;
  pline 1.8
    to 2.3 tgrey, to (2.4,+0), to (2.6,+1.5), to (2.7,+0) ahe,
    to 3.3 tgrey, to (3.4,+0), to (3.6,+1.5), to (3.7,+0) ahe,
    to 4.3 tgrey, to (4.4,+0), to (4.6,+1.5), to (4.7,+0) ahe,
    to 5.3 tgrey, to (5.4,+0), to (5.6,+1.5), to (5.7,+0) ahe,
    to 6.3 tgrey;

object 2;
  pline (1.8,+10) to 2.3 tgrey;

object 3;
  pline (3.3,+2)
    to 2.85 tgrey, to (2.7,+0) tgrey,
    to (2.5,+0), to (2.25,+1.5), to (2.0,+0) ahe;
  text seg 2 centre "408";

object 4;
  pline (4.3,+1.5)
    to 3.9 tgrey, to (3.7,+0) tgrey,
    to (3.5,+0), to (3.3,+1.5), to (3.1,+0) ahe,
    to 2.9 tgrey, to (2.7,+0) tgrey,
    to (2.5,+0), to (2.25,+1.5), to (2.0,+0) ahe;
  text seg 2 centre "408";
  text seg 7 centre "408";

object 5;
  pline (5.3,+1.5)
    to 4.9 tgrey, to (4.7,+0) tgrey,
to (4.5,+0), to (4.3,+1.5), to (4.1,+0) ahe,
to 3.9 tgrey, to (3.7,+0) tgrey,
to (3.5,+0), to (3.3,+1.5), to (3.1,+0) ahe,
to 2.9 tgrey, to (2.7,+0) tgrey,
to (2.5,+0), to (2.25,+1.5), to (2.0,+0) ahe;
text seg 2 centre "408";
text seg 7 centre "408";
text seg 12 centre "408";

object 6;
pline (6.3,+1.5)
to 5.9 tgrey, to (5.7,+0) tgrey,
to (5.5,+0), to (5.3,+1.5), to (5.1,+0) ahe;
to 4.9 tgrey, to (4.7,+0) tgrey,
to (4.5,+0), to (4.3,+1.5), to (4.1,+0) ahe;
to 3.9 tgrey, to (3.7,+0) tgrey,
to (3.5,+0), to (3.3,+1.5), to (3.1,+0) ahe;
to 2.9 tgrey, to (2.7,+0) tgrey,
to (2.5,+0), to (2.25,+1.5), to (2.0,+0) ahe;
text seg 2 centre "408";
text seg 7 centre "408";
text seg 12 centre "408";
text seg 17 centre "408";

object 7:
pline (1.63,4.1) to (1.73,+0);

object 8;
pline (1.68,4.1) to (+0,14) arrowhead at end;
text centre "64*T1";

object 9;
pline (1.2,13.1) to (1.5,+0) color "white";
text centre "(timeout)";

6. IANA Considerations

This document does not create a new registry nor does it register any values in existing registries; no IANA action is required.

7. Acknowledgements

Thanks to the RSE and the Design Team members for their helpful comments and suggestions for SVG 1.2 RFC.
8. Revision History [RFC Editor please delete]

version -13, 16 Oct 15:
 Added Informative Reference to XML2RFC v3 draft on 'how to include
SVG diagrams'.
 I Added nformative Reference to Nevil's home page for a version of
 this I-D with its SVG diagrams included.
 Added Informative References to RFC0793 and I-Ds for the example
 diagrams.
 Changed 'colour' to 'color' so as to be consistent.
 Fixed other typos (thanks to Dave Thaler for all these)!
 Removed 'grey' color from example diagrams.

version -12, 24 Sep 15:
 Appendix A added: a complete relax-ng compact (rnc) schema for
 SVG-1.2-RFC.
 Section 2.1: Elements/attributes/properties table updated to match
 the schema in Appendix A.

version -11, 17 Aug 15:
 Section 1: Fixed typo in "Details are expected to change"
 paragraph.

version -10, 14 Aug 15:
 Section 1: Added "Details are expected to change" paragraph.

version -09, 31 Mar 15:
 No changes, version number incremented to keep draft alive

version -08, 29 Sep 14:
 Section 1: Changed comment about diagrams 'not being normative' to
 'not complete specifications in themselves.
 Section 2.1: Added SVG 1.2 Tiny 'id' attribute because most
drawing packages use it in constructing drawings.
 Section 2.1: Added SVG 1.2 Tiny 'role' attribute so that ARIA can
 use it.
 Section 3: added comment about changes to xml2rfc required to
 include SVG diagrams.
 Section 4: Added reference to svg-aria-primer.

version -07, 3 Jul 14:
 Expanded text about Accessibility in 'how to create SVG drawings'
 section into 'Accessibility Considerations section. Added two SVG
 Accessibility references to support that.

version -06, 26 Jun 14:
 Remove trailing / from URL in section 4; the html version on
 tools.ietf.org/html assumed the next word was part of that URL.
version -05, 25 Jun 14:
Improved section on ‘how to create SVG drawings’ By adding some
text about which elements aren’t allowed in SVG 1.2 RFC.
Added more text describing the tool for checking, stripping out or
replacing incompatible elements and attributes from an SVG file.

version -04, 30 Apr 14:
Fixed typos, used full references for two of the w3c refs - each
had an author name using UTF8 characters.
Moved the Elements and Attributes appendix up earlier to make it
sub-section 2.1.
Disclaimer added to the Meta-languages section.

version -03, 14 Apr 14:
Added two more example diagrams; a simple packet layout, and a
diagram that uses lots of diagonal lines.

version -02, 12 Feb 14:
Added metalanguage example to make time-sequence drawings.

version -01, 11 Feb 14:
Allow links to ‘long-term stable URIs’
Link URIs must be ASCII only
Need for tools to check SVG 1.2 RFC compatibility and to strip
‘unnecessary’ attributes explicitly stated.
Statement that drawings can’t be normative removed; Postscript-
only RFCs already exist.
Added most attributes and elements to the Appendix.

version -00, 29 Jan 14:
Initial version, using content from Nevil’s
emails to the Design Team.

9. References

9.1. Normative References

[RF6949] Flanagan, H. and N. Brownlee, "RFC Series Format
Requirements and Future Development", RFC 6949,
DOI 10.17487/RFC6949, May 2013,
9.2. Informative References

[W3C.REC-SVGTiny12-20081222]

[W3C.REC-CSS2-20110607]

[SVG-ACCESS-TIPS]

[SVG-ARIA-PRIMER]

[SVG-USING-ARIA]

/XML2RFCv3]
Appendix A. RELAX NG Compact (rnc) Schema for SVG 1.2 RFC

The following rnc schema can be used to check whether an svg file conforms to SVG 1.2 RFC. For example, if this schema were contained in a file called SVG-1.2-RFC.rnc, the following command will test whether svg file diagram.svg is a conformant SVG 1.2 RFC drawing.

```
jing -c SVG-1.2-RFC.rnc diagram.svg
```

---
SVG 1.2 RFC rnc schema; Nevil Brownlee, Thu 24 Sep 2015 (NZST)

```
default namespace = "http://www.w3.org/2000/svg"
namespace ns1 = "http://www.w3.org/1999/xlink"

rfc-color = (  # SVG-1.2-RFC doesn’t allow "color or grey-scale"
  "black" | "white" | "#000000" | "#FFFFFF" | "#ffffff" | "inherit"
)

start = svg

svg =
  element svg {
    (attribute fill-opacity { "inherit" | xsd:string }?,
     attribute stroke-opacity { "inherit" | xsd:string }?)
& (attribute fill { "none" | rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke { rfc-color }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap {
```

Brownlee & IAB  Expires April 18, 2016
"butt" | "round" | "square" | "inherit"
),
attribute stroke-linejoin {
   "miter" | "round" | "bevel" | "inherit"
},
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering {
   "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
})?
& attribute vector-effect {
   "none" | "non-scaling-stroke" | "inherit"
}?
& {attribute direction { "ltr" | "rtl" | "inherit" }?,
   attribute unicode-bidi {
      "normal" | "embed" | "bidi-override" | "inherit"
}?
& {attribute solid-color { rfc-color }?,
   attribute solid-opacity { "inherit" | xsd:string }?
} & {attribute display-align {
   "auto" | "before" | "center" | "after" | "inherit"
}?,
   attribute line-increment { "auto" | "inherit" | xsd:string }?
} & {attribute stop-color { rfc-color }?,
   attribute stop-opacity { "inherit" | xsd:string }?
} & {attribute font-family { "inherit" | xsd:string }?,
   attribute font-size { "inherit" | xsd:string }?,
   attribute font-style {
      "normal" | "italic" | "oblique" | "inherit"
}?,
   attribute font-variant { "normal" | "small-caps" | "inherit" }?,
   attribute font-weight {
      "normal"
      | "bold"
      | "bolder"
      | "lighter"
}?,
   attribute text-anchor {
      "start" | "middle" | "end" | "inherit"
}?,
   attribute text-align {
      "start" | "center" | "end" | "inherit"
}?
})
(attribute id { xsd:NCName }
 | attribute xml:id { xsd:NCName })?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute datatype { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,

attribute width { xsd:string }?,
attribute height { xsd:string }?,
attribute preserveAspectRatio {
  xsd:string { pattern = "\s*(none|xMidYMid)\s*(meet)\s*\s*" } }?,
attribute viewBox { text }?,
attribute zoomAndPan { "disable" | "magnify" }?,
attribute version {
  xsd:string "1.0" | xsd:string "1.1" | xsd:string "1.2"
}?,
attribute baseProfile {
  xsd:string "none"
  | xsd:string "tiny"
  | xsd:string "basic"
  | xsd:string "full"
}?,
attribute contentScriptType { xsd:string }?,
attribute snapshotTime { xsd:string "none" | xsd:string }?,
attribute timelineBegin {
  xsd:string "onLoad" | xsd:string "onStart"
}?,
attribute playbackOrder {
  xsd:string "all" | xsd:string "forwardOnly"
}?,

desc
  title
  path
  rect
  circle
  line
  ellipse
  polyline
  polygon
  solidColor
desc =
  desc {
    (attribute desc { xsd:NCName }
      | attribute xml:id { xsd:NCName })?,
    attribute xml:base { xsd:anyURI | xsd:string }?,
    attribute xml:lang { xsd:language }?,
    attribute class { xsd:NMTOKENS }?,
    attribute role { xsd:string }?,
    attribute rel { xsd:string }?,
    attribute rev { xsd:string }?,
    attribute typeof { xsd:string }?,
    attribute content { xsd:string }?,
    attribute datatype { xsd:string }?,
    attribute resource { xsd:string }?,
    attribute about { xsd:string }?,
    attribute property { xsd:string }?,
    attribute xml:space { "default" | "preserve" }?,
    attribute requiredFeatures { xsd:string }?,
    attribute requiredExtensions { xsd:string }?,
    attribute requiredFormats { xsd:string }?,
    attribute requiredFonts { xsd:string }?,
    attribute systemLanguage { xsd:string }?,
    ((attribute display {
      "inline"
      | "block"
      | "list-item"
      | "run-in"
      | "compact"
      | "marker"
      | "table"
      | "inline-table"
      | "table-row-group"
      | "table-header-group"
      | "table-footer-group"
      | "table-row"
      | "table-column-group"
      | "table-column"
      | "table-cell"
      | "table-caption"
    })*)
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
((attribute display {
  "inline"
  "block"
  "list-item"
  "run-in"
  "compact"
  "marker"
  "table"
  "inline-table"
  "table-row-group"
  "table-header-group"
  "table-footer-group"
  "table-row"
  "table-column-group"
  "table-column"
  "table-cell"
  "table-caption"
  "none"
  "inherit"
}),
attribute visibility { "visible" | "hidden" | "collapse" | "inherit" }?,
attribute image-rendering {
  "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
}),
attribute shape-rendering {
  "auto"
  "optimizeSpeed"
  "crispEdges"
  "geometricPrecision"
  "inherit"
}),
attribute text-rendering {
  "auto"
  "optimizeSpeed"
  "optimizeLegibility"
  "geometricPrecision"
  "inherit"
}),
attribute buffered-rendering {
  "auto" | "dynamic" | "static" | "inherit"
}& {attribute viewport-fill { "none" | rfc-color }?,
attribute viewport-fill-opacity { "inherit" | xsd:string }?),
text
}
path =
element path {
  (attribute id { xsd:NCName } |
    attribute xml:id { xsd:NCName })?,
  attribute xml:base { xsd:anyURI | xsd:string }?,
  attribute xml:lang { xsd:language? }?,
  attribute class { xsd:NMTOKENS }?,
  attribute role { xsd:string }?,
  attribute rel { xsd:string }?,
  attribute rev { xsd:string }?,
  attribute typeof { xsd:string }?,
  attribute content { xsd:string }?,
  attribute datatype { xsd:string }?,
  attribute resource { xsd:string }?,
  attribute about { xsd:string }?,
  attribute property { xsd:string }?,
  attribute xml:space { "default" | "preserve" }?,
  attribute transform { xsd:string | "none" }?,
  ((attribute fill-opacity { "inherit" | xsd:string }?,
    attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "none" | rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke { rfc-color }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap {
      "butt" | "round" | "square" | "inherit"
    }?,
    attribute stroke-linejoin {
      "miter" | "round" | "bevel" | "inherit"
    }?,
    attribute stroke-miterlimit { "inherit" | xsd:string }?,
    attribute stroke-width { "inherit" | xsd:string }?,
    attribute color { rfc-color }?,
    attribute color-rendering {
      "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
    }?)
  & attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit"
  }?
  & (attribute direction { "ltr" | "rtl" | "inherit" }?,
    attribute unicode-bidi {
      "normal" | "embed" | "bidi-override" | "inherit"
    })
  & (attribute solid-color { rfc-color }?,
    attribute solid-opacity { "inherit" | xsd:string }?)
  & (attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
  }?,

attribute line-increment { "auto" | "inherit" | xsd:string }?,
& (attribute stop-color { rfc-color }?,
  attribute stop-opacity { "inherit" | xsd:string }?,
& (attribute font-family { "inherit" | xsd:string }?,
  attribute font-size { "inherit" | xsd:string }?,
  attribute font-style { "normal" | "italic" | "oblique" | "inherit" }?,
  attribute font-variant { "normal" | "small-caps" | "inherit" }?,
  attribute font-weight { "normal" | "bold" | "bolder" | "lighter" | "inherit" }?),
attribute text-anchor { "start" | "middle" | "end" | "inherit" }?,
attribute text-align { "start" | "center" | "end" | "inherit" }?),
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute d { xsd:string }?,
attribute pathLength { xsd:string }?,
attribute style { xsd:string }?, # Added to SVG-1.2-RFC (Inkscape)
(desc | title)*
}
rect =
element rect {
  (attribute id { xsd:NCName }
   | attribute xml:id { xsd:NCName }?),
  attribute xml:base { xsd:anyURI | xsd:string }?,
  attribute xml:lang { xsd:language? }?,
  attribute class { xsd:NMTOKENS }?,
  attribute role { xsd:string }?,
  attribute rel { xsd:string }?,
  attribute rev { xsd:string }?,
  attribute typeof { xsd:string }?,
  attribute content { xsd:string }?,
  attribute datatype { xsd:string }?,
  attribute resource { xsd:string }?,
  attribute about { xsd:string }?,
  attribute requiredFeatures { xsd:string }?,
  attribute requiredExtensions { xsd:string }?,
  attribute requiredFormats { xsd:string }?,
  attribute requiredFonts { xsd:string }?,
  attribute systemLanguage { xsd:string }?,
  attribute d { xsd:string }?,
  attribute pathLength { xsd:string }?,
  attribute style { xsd:string }?, # Added to SVG-1.2-RFC (Inkscape)
  (desc | title)*
}
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute transform { xsd:string | "none" }?,
(attribute fill-opacity { "inherit" | xsd:string }?,
attribute stroke-opacity { "inherit" | xsd:string }?)
& (attribute fill { "none" | rfc-color }?,
attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
attribute stroke { rfc-color }?,
attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
attribute stroke-dashoffset { "inherit" | xsd:string }?,
attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
& attribute vector-effect { "none" | "non-scaling-stroke" | "inherit" }?
& (attribute direction { "ltr" | "rtl" | "inherit" }?,
attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?)
& (attribute solid-color { rfc-color }?,
attribute solid-opacity { "inherit" | xsd:string }?)
& (attribute display-align { "auto" | "before" | "center" | "after" | "inherit" }?,
attribute line-increment { "auto" | "inherit" | xsd:string }?)
& (attribute stop-color { rfc-color }?,
attribute stop-opacity { "inherit" | xsd:string }?)
& (attribute font-family { "inherit" | xsd:string }?,
attribute font-size { "inherit" | xsd:string }?,
attribute font-style { "normal" | "italic" | "oblique" | "inherit" }?,
attribute font-variant { "normal" | "small-caps" | "inherit" }?,
attribute font-weight { "normal" | "bold" | "bolder" | "lighter" }
attribute text-anchor {
  "start" | "middle" | "end" | "inherit"
},
attribute text-align {
  "start" | "center" | "end" | "inherit"
}),
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
x attribute x { xsd:string }?,
x attribute y { xsd:string }?,
x attribute height { xsd:string }?,
x attribute rx { xsd:string }?,
x attribute ry { xsd:string }?,
x attribute style { xsd:string }?, # Added to SVG-1.2-RFC (Inkscape)
  (desc | title)*
} circle =
  element circle {
    (attribute id { xsd:NCName } |
    attribute xml:id { xsd:NCName })?,
x attribute xml:base { xsd:anyURI | xsd:string }?,
x attribute xml:lang { xsd:language? }?,
x attribute class { xsd:NMTOKENS }?,
x attribute role { xsd:string }?,
x attribute rel { xsd:string }?,
x attribute rev { xsd:string }?,
x attribute typeof { xsd:string }?,
x attribute content { xsd:string }?,
x attribute datatype { xsd:string }?,
x attribute resource { xsd:string }?,
x attribute about { xsd:string }?,
x attribute property { xsd:string }?,
x attribute xml:space { "default" | "preserve" }?,
x attribute transform { xsd:string | "none" }?,
  ((attribute fill-opacity { "inherit" | xsd:string }?,
x attribute stroke-opacity { "inherit" | xsd:string }?))
x & {attribute fill { "none" | rfc-color }?,
x attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
x attribute stroke { rfc-color }?,
x attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
x attribute stroke-dashoffset { "inherit" | xsd:string }?,

Brownlee & IAB Expires April 18, 2016 [Page 26]
attribute stroke-linecap {
    "butt" | "round" | "square" | "inherit"
}\?,
attribute stroke-linejoin {
    "miter" | "round" | "bevel" | "inherit"
}\?,
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?),
attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit"
}\?,
& attribute direction { "ltr" | "rtl" | "inherit" }?,
attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?),
& attribute solid-color { rfc-color }?,
attribute solid-opacity { "inherit" | xsd:string }?
& attribute display-align { "auto" | "before" | "center" | "after" | "inherit" }?,
attribute line-increment { "auto" | "inherit" | xsd:string }?
& (attribute stop-color { rfc-color }?),
attribute stop-opacity { "inherit" | xsd:string }?
& (attribute font-family { "inherit" | xsd:string }?,
attribute font-size { "inherit" | xsd:string }?,
attribute font-style { "normal" | "italic" | "oblique" | "inherit" }?,
attribute font-variant { "normal" | "small-caps" | "inherit" }?,
attribute font-weight { "normal"
    | "bold"
    | "bolder"
    | "lighter"
    | "inherit"
    }?),
attribute text-anchor { "start" | "middle" | "end" | "inherit" }?,
attribute text-align { "start" | "center" | "end" | "inherit" }?),
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,

Brownlee & IAB Expires April 18, 2016 [Page 27]
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute cx { xsd:string }?,
attribute cy { xsd:string }?,
attribute r { xsd:string }?,
attribute style { xsd:string }?,  # Added to SVG-1.2-RFC (Inkscape)
(desc
  | title)*
}
line =
  element line {
    (attribute id { xsd:NCName }
        | attribute xml:id { xsd:NCName } )?,
    attribute xml:base { xsd:anyURI | xsd:string }?,
    attribute xml:lang { xsd:language }?,
    attribute class { xsd:NMTOKENS }?,
    attribute role { xsd:string }?,
    attribute rel { xsd:string }?,
    attribute rev { xsd:string }?,
    attribute typeof { xsd:string }?,
    attribute content { xsd:string }?,
    attribute datatype { xsd:string }?,
    attribute resource { xsd:string }?,
    attribute about { xsd:string }?,
    attribute property { xsd:string }?,
    attribute xml:space { "default" | "preserve" }?,
    attribute transform { xsd:string | "none" }?,
    ((attribute fill-opacity { "inherit" | xsd:string }?,
      attribute stroke-opacity { "inherit" | xsd:string }?)
      & (attribute fill { "none" | rfc-color }?,
        attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
        attribute stroke { rfc-color }?),
      attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
      attribute stroke-dashoffset { "inherit" | xsd:string }?,
      attribute stroke-linecap {"butt" | "round" | "square" | "inherit" }?,
      attribute stroke-linejoin {"miter" | "round" | "bevel" | "inherit" }?,
      attribute stroke-miterlimit { "inherit" | xsd:string }?,
      attribute stroke-width { "inherit" | xsd:string }?,
      attribute color { rfc-color }?,
      attribute color-rendering {"auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
    & attribute vector-effect {
ellipse =
element ellipse {
    (attribute id { xsd:NCName })
attribute xml:id { xsd:NCName }?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute datatype { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute transform { xsd:string | "none" }?,
(attribute fill-opacity { "inherit" | xsd:string }?,
attribute stroke-opacity { "inherit" | xsd:string }?),
attribute vector-effect { "none" | "non-scaling-stroke" | "inherit" }?
(attribute fill-rule { "inheri" | "nonzero" | "evenodd" }?,
attribute stroke { rfc-color }?,
attribute stroke-dasharray { "inheri" | "none" | xsd:string }?,
attribute stroke-dashoffset { "inheri" | xsd:string }?,
attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?
(attribute direction { "ltr" | "rtl" | "inherit" }?,
attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?),
(attribute solid-color { rfc-color }?,
attribute solid-opacity { "inherit" | xsd:string }?)
(attribute display-align { "auto" | "before" | "center" | "after" | "inherit" }?,
attribute line-increment { "auto" | "inherit" | xsd:string }?)
(attribute stop-color { rfc-color }?,

attribute stop-opacity { "inherit" | xsd:string }?,
& (attribute font-family { "inherit" | xsd:string }?,
  attribute font-size { "inherit" | xsd:string }?,
  attribute font-style {
    "normal" | "italic" | "oblique" | "inherit"
  }?,
  attribute font-variant { "normal" | "small-caps" | "inherit" }?,
  attribute font-weight {
    "normal"
    | "bold"
    | "bolder"
    | "lighter"
    | "inherit"
  }?,
  attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
  }?,
  attribute text-align {
    "start" | "center" | "end" | "inherit"
  }?),
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute rx { xsd:string }?,
attribute ry { xsd:string }?,
attribute cx { xsd:string }?,
attribute cy { xsd:string }?,
attribute style { xsd:string }?,  # Added to SVG-1.2-RFC (Inkscape)
(desc | title)*
)

polyline =
element polyline {
  (attribute id { xsd:NCName }
   | attribute xml:id { xsd:NCName })?,
  attribute xml:base { xsd:anyURI | xsd:string }?,
  attribute xml:lang { xsd:language? }?,
  attribute class { xsd:NMTOKENS }?,
  attribute role { xsd:string }?,
  attribute rel { xsd:string }?,
  attribute typeof { xsd:string }?,
  attribute content { xsd:string }?,
  attribute datatype { xsd:string }?,
  attribute resource { xsd:string }?,
  attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute transform { xsd:string | "none" }?,
{(attribute fill-opacity { "inherit" | xsd:string }?,
    attribute stroke-opacity { "inherit" | xsd:string }?)
& {attribute fill { "none" | rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke { rfc-color }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
    attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
    attribute stroke-miterlimit { "inherit" | xsd:string }?,
    attribute stroke-width { "inherit" | xsd:string }?,
    attribute color { rfc-color }?,
    attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
& attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit" }?
& {attribute direction { "ltr" | "rtl" | "inherit" }?,
    attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?)
& {attribute solid-color { rfc-color }?,
    attribute solid-opacity { "inherit" | xsd:string }?)
& {attribute display-align { "auto" | "before" | "center" | "after" | "inherit" }?,
    attribute line-increment { "auto" | "inherit" | xsd:string }?)
& {attribute stop-color { rfc-color }?,
    attribute stop-opacity { "inherit" | xsd:string }?)
& {attribute font-family { "inherit" | xsd:string }?,
    attribute font-size { "inherit" | xsd:string }?,
    attribute font-style { "normal" | "italic" | "oblique" | "inherit" }?,
    attribute font-variant { "normal" | "small-caps" | "inherit" }?,
    attribute font-weight { "normal" | "bold" | "bolder" | "lighter" }?
attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
}),
attribute text-align {
    "start" | "center" | "end" | "inherit"
}),
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute points { xsd:string }?,
(desc
title)*
}
polygon =
element polygon {
    (attribute id { xsd:NCName } |
        attribute xml:id { xsd:NCName })?,
    attribute xml:base { xsd:anyURI | xsd:string }?,
    attribute xml:lang { xsd:language? }?,
    attribute class { xsd:NMTOKENS }?,
    attribute role { xsd:string }?,
    attribute rel { xsd:string }?,
    attribute rev { xsd:string }?,
    attribute typeof { xsd:string }?,
    attribute content { xsd:string }?,
    attribute datatype { xsd:string }?,
    attribute resource { xsd:string }?,
    attribute about { xsd:string }?,
    attribute property { xsd:string }?,
    attribute xml:space { "default" | "preserve" }?,
    attribute transform { xsd:string | "none" }?,
    ((attribute fill-opacity { "inherit" | xsd:string }?,
        attribute stroke-opacity { "inherit" | xsd:string }?)
& (attribute fill { "none" | rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke { rfc-color }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap {
        "butt" | "round" | "square" | "inherit"
    }?,
    attribute stroke-linejoin {
        "miter" | "round" | "bevel" | "inherit"
    }?,
    ...
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering {
    "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
}?
& attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit"
}?
& (attribute direction { "ltr" | "rtl" | "inherit" }?,
attribute unicode-bidi {
    "normal" | "embed" | "bidi-override" | "inherit"
})?
& (attribute solid-color { rfc-color }?,
attribute solid-opacity { "inherit" | xsd:string }?)
& (attribute display-align { "auto" | "before" | "center" | "after" | "inherit"
}?,
attribute line-increment { "auto" | "inherit" | xsd:string }?)
& (attribute stop-color { rfc-color }?,
attribute stop-opacity { "inherit" | xsd:string }?)
& (attribute font-family { "inherit" | xsd:string }?,
attribute font-size { "inherit" | xsd:string }?,
attribute font-style { "normal" | "italic" | "oblique" | "inherit"
}?,
attribute font-variant { "normal" | "small-caps" | "inherit" }?,
attribute font-weight { "normal" |
    "bold"
    "bolder"
    "lighter"
    "inherit"
}?,
attribute text-anchor { "start" | "middle" | "end" | "inherit"
}?,
attribute text-align { "start" | "center" | "end" | "inherit"
}?)
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute points { xsd:string }?,
attribute style { xsd:string }?,  # Added to SVG-1.2-RFC (Inkscape)
(desc
solidColor =
  element solidColor {
    ((attribute fill-opacity { "inherit" | xsd:string }?,
      attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "none" | rfc-color }?,
     attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
     attribute stroke { rfc-color }?,
     attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
     attribute stroke-dashoffset { "inherit" | xsd:string }?,
     attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
     attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
     attribute stroke-miterlimit { "inherit" | xsd:string }?,
     attribute stroke-width { "inherit" | xsd:string }?,
     attribute color { rfc-color }?,
     attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
  & attribute vector-effect { "none" | "non-scaling-stroke" | "inherit" }?
  & {attribute direction { "ltr" | "rtl" | "inherit" }?,
     attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?)
  & {attribute solid-color { rfc-color }?,
     attribute solid-opacity { "inherit" | xsd:string }?)
  & (attribute display-align { "auto" | "before" | "center" | "after" | "inherit" }?,
     attribute line-increment { "auto" | "inherit" | xsd:string }?)
  & {attribute stop-color { rfc-color }?,
     attribute stop-opacity { "inherit" | xsd:string }?)
  & {attribute font-family { "inherit" | xsd:string }?,
     attribute font-size { "inherit" | xsd:string }?,
     attribute font-style { "normal" | "italic" | "oblique" | "inherit" }?,
     attribute font-variant { "normal" | "small-caps" | "inherit" }?,
     attribute font-weight { "normal" | "bold" | "bolder" }?
attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
},
attribute text-align {
    "start" | "center" | "end" | "inherit"
}), (attribute id { xsd:NCName }
    attribute xml:id { xsd:NCName })*?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute datatype { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
(desc
title)*
}
textArea =
element textArea {
    ((attribute fill-opacity { "inherit" | xsd:string }?),
    attribute stroke-opacity { "inherit" | xsd:string }?)
& (attribute fill { "none" | rfc-color }?,
    attribute stroke { rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap {"butt" | "round" | "square" | "inherit" }?,
    attribute stroke-linejoin {"miter" | "round" | "bevel" | "inherit" }?,
    attribute stroke-miterlimit { "inherit" | xsd:string }?,
    attribute stroke-width { "inherit" | xsd:string }?,
    attribute color { rfc-color }?,
    attribute color-rendering {"auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
}
& attribute vector-effect {
  "none" | "non-scaling-stroke" | "inherit"
}?
& {attribute direction { "ltr" | "rtl" | "inherit" }?,
  attribute unicode-bidi {
    "normal" | "embed" | "bidi-override" | "inherit"
}?}
& {attribute solid-color { rfc-color }?,
  attribute solid-opacity { "inherit" | xsd:string }?}
& {attribute display-align {
  "auto" | "before" | "center" | "after" | "inherit"
}?,
  attribute line-increment { "auto" | "inherit" | xsd:string }?}
& {attribute stop-color { rfc-color }?,
  attribute stop-opacity { "inherit" | xsd:string }?}
& {attribute font-family { "inherit" | xsd:string }?,
  attribute font-size { "inherit" | xsd:string }?,
  attribute font-style { "normal" | "italic" | "oblique" | "inherit"
}?,
  attribute font-variant { "normal" | "small-caps" | "inherit" }?,
  attribute font-weight {
    "normal"
    | "bold"
    | "bolder"
    | "lighter"
    | "inherit"
}?,
  attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
}?,
  attribute text-align {
    "start" | "center" | "end" | "inherit"
}?)
(attribute id { xsd:NCName })
| attribute xml:id { xsd:NCName }?),
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute datatype { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,

attribute xml:space { "default" | "preserve" }?,
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute transform { xsd:string | "none" }?,
attribute x { xsd:string }?,
attribute y { xsd:string }?,
attribute width { xsd:string | "auto" }?,
attribute height { xsd:string | "auto" }?,
(tspan
desc
title
tspan_2
text
  a_2)+
}
linearGradient =
element linearGradient {
  ((attribute fill-opacity { "inherit" | xsd:string }?,
    attribute stroke-opacity { "inherit" | xsd:string }?)
   & (attribute fill { "none" | rfc-color }?,
     attribute stroke { rfc-color }?,
     attribute fill-opacity { "inherit" | xsd:string }?,
     attribute stroke-opacity { "inherit" | xsd:string }?,
     attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
     attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
     attribute stroke-dashoffset { "inherit" | xsd:string }?,
     attribute stroke-linecap {
       "butt" | "round" | "square" | "inherit"
     }?,
     attribute stroke-linejoin {
       "miter" | "round" | "bevel" | "inherit"
     }?,
     attribute stroke-miterlimit { "inherit" | xsd:string }?,
     attribute stroke-width { "inherit" | xsd:string }?,
     attribute color { rfc-color }?,
     attribute color-rendering {
       "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
     }?
   & attribute vector-effect {
     "none" | "non-scaling-stroke" | "inherit"
   }?
   & (attribute direction { "ltr" | "rtl" | "inherit" }?,
     attribute unicode-bidi {
       "normal" | "embed" | "bidi-override" | "inherit"
     }?)
   & (attribute solid-color { rfc-color }?,
     attribute solid-opacity { "inherit" | xsd:string }?)
   )
}
& (attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
}),
attribute line-increment ( "auto" | "inherit" | xsd:string )?,
& (attribute stop-color ( rfc-color )?,
attribute stop-opacity { "inherit" | xsd:string }?)?
& (attribute font-family { "inherit" | xsd:string }?,
attribute font-size { "inherit" | xsd:string }?,
attribute font-style {
    "normal" | "italic" | "oblique" | "inherit"
},
attribute font-variant { "normal" | "small-caps" | "inherit" }?,
attribute font-weight {
    "normal"
    | "bold"
    | "bolder"
    | "lighter"
    | "inherit"
},
attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
}),
attribute text-align {
    "start" | "center" | "end" | "inherit"
}),
attribute gradientUnits { "userSpaceOnUse" | "objectBoundingBox" }?,
(attribute id { xsd:NCName }
  | attribute xml:id { xsd:NCName })),
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute x1 { xsd:string }?,
attribute y1 { xsd:string }?,
attribute x2 { xsd:string }?,
attribute y2 { xsd:string }?,
(desc |
  title)*}
radialGradient =
element radialGradient {
  ((attribute fill-opacity { "inherit" | xsd:string })?,
   attribute stroke-opacity { "inherit" | xsd:string })?)
  & (attribute fill { "none" | rfc-color }?,
   attribute fill-rule { "inherit" | "nonzero" | "evenodd" })?,
  attribute stroke { rfc-color }?,
  attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
  attribute stroke-dashoffset { "inherit" | xsd:string }?,
  attribute stroke-linecap {
    "butt" | "round" | "square" | "inherit"
  }?,
  attribute stroke-linejoin {
    "miter" | "round" | "bevel" | "inherit"
  }?,
  attribute stroke-miterlimit { "inherit" | xsd:string }?,
  attribute stroke-width { "inherit" | xsd:string }?,
  attribute color { rfc-color }?,
  attribute color-rendering {
    "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
  }?)
  & attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit"
  }?
  & (attribute direction { "ltr" | "rtl" | "inherit" })?,
  attribute unicode-bidi {
    "normal" | "embed" | "bidi-override" | "inherit"
  }?)
  & (attribute solid-color { rfc-color }?),
  attribute solid-opacity { "inherit" | xsd:string }?)
  & (attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
  }?,
    attribute line-increment { "auto" | "inherit" | xsd:string }?)
  & (attribute stop-color { rfc-color }?),
  attribute stop-opacity { "inherit" | xsd:string }?)
  & (attribute font-family { "inherit" | xsd:string })?,
  attribute font-size { "inherit" | xsd:string }?,
  attribute font-style { 
    "normal" | "italic" | "oblique" | "inherit"
  }?,
  attribute font-variant { "normal" | "small-caps" | "inherit" }?,
  attribute font-weight { 
    "normal"
    "bold"
    "bolder"
    "lighter"
    "inherit"}
attribute text-anchor {
  "start" | "middle" | "end" | "inherit"
},
attribute text-align {
  "start" | "center" | "end" | "inherit"
}),
attribute gradientUnits { "userSpaceOnUse" | "objectBoundingBox" }?,
(attribute id { xsd:NCName }
 | attribute xml:id { xsd:NCName })?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role ( xsd:string )?,
attribute rel ( xsd:string )?,
attribute rev ( xsd:string )?,
attribute typeof ( xsd:string )?,
attribute content ( xsd:string )?,
attribute datatype ( xsd:string )?,
attribute resource ( xsd:string )?,
attribute about ( xsd:string )?,
attribute property ( xsd:string )?,
attribute xml:space { "default" | "preserve" }?,
attribute cx { xsd:string }?,
attribute cy { xsd:string }?,
attribute r { xsd:string }?,
(desc
 | title)*
)
\text =

element text {
  ((attribute fill-opacity { "inherit" | xsd:string }?),
  attribute stroke-opacity { "inherit" | xsd:string })
  & (attribute fill { "none" | rfc-color }?,
  attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
  attribute stroke { rfc-color }?,
  attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
  attribute stroke-dashoffset { "inherit" | xsd:string }?,
  attribute stroke-linecap { "butt" | "round" | "square" | "inherit"
  }?,
  attribute stroke-linejoin { 
    "miter" | "round" | "bevel" | "inherit"
  }?,
  attribute stroke-miterlimit { "inherit" | xsd:string }?,
  attribute stroke-width { "inherit" | xsd:string }?,
  attribute color { rfc-color }?,
  attribute color-rendering {
"auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"

& attribute vector-effect {
   "none" | "non-scaling-stroke" | "inherit"

& (attribute direction { "ltr" | "rtl" | "inherit" }?,
   attribute unicode-bidi {
      "normal" | "embed" | "bidi-override" | "inherit"
   })?

& (attribute solid-color { rfc-color }?,
   attribute solid-opacity { "inherit" | xsd:string }?)

& (attribute display-align {
   "auto" | "before" | "center" | "after" | "inherit"
   },
   attribute line-increment { "auto" | "inherit" | xsd:string }?)

& (attribute stop-color { rfc-color }?,
   attribute stop-opacity { "inherit" | xsd:string }?)

& (attribute font-family { "inherit" | xsd:string }?,
   attribute font-size { "inherit" | xsd:string }?,
   attribute font-style {
      "normal" | "italic" | "oblique" | "inherit"
   }?,
   attribute font-variant { "normal" | "small-caps" | "inherit" }?,
   attribute font-weight {
      "normal"
      | "bold"
      | "bolder"
      | "lighter"
      | "inherit"
   }?,
   attribute text-anchor {
      "start" | "middle" | "end" | "inherit"
   }?,
   attribute text-align {
      "start" | "center" | "end" | "inherit"
   })

(attribute id { xsd:NCName } |
   attribute xml:id { xsd:NCName }?)?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute resource { xsd:string }?,

attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute transform { xsd:string | "none" }?,
attribute x { xsd:string }?,
attribute y { xsd:string }?,
attribute rotate { xsd:string }?,
attribute style { xsd:string }?,  # Added to SVG-1.2-RFC (Inkscape)
(desc
title
tspan_2
text
t_2)+
}
g =
element g {
  ((attribute fill-opacity { "inherit" | xsd:string }?,
    attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "none" | rfc-color }?,
    attribute stroke { rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
    attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
    attribute stroke-miterlimit { "inherit" | xsd:string }?,
    attribute stroke-width { "inherit" | xsd:string }?,
    attribute color { rfc-color }?,
    attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
  & attribute vector-effect { "none" | "non-scaling-stroke" | "inherit" }?
  & (attribute direction { "ltr" | "rtl" | "inherit" }?),
  attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?)
  & (attribute solid-color { rfc-color }?)
}
attribute solid-opacity { "inherit" | xsd:string }?

& (attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
})?,

attribute line-increment ( "auto" | "inherit" | xsd:string )?

& (attribute stop-color { rfc-color }?),

attribute stop-opacity { "inherit" | xsd:string }?

& (attribute font-family { "inherit" | xsd:string }?),

attribute font-size { "inherit" | xsd:string }?,

attribute font-style {
    "normal" | "italic" | "oblique" | "inherit"
}?,

attribute font-variant { "normal" | "small-caps" | "inherit" }?,

attribute font-weight {
    "normal"
    "bold"
    "bolder"
    "lighter"
    "inherit"
}?,

attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
}?,

attribute text-align {
    "start" | "center" | "end" | "inherit"
}?,

(attribute id { xsd:NCName }
  | attribute xml:id { xsd:NCName })?,

attribute xml:base { xsd:anyURI | xsd:string }?,

attribute xml:lang { xsd:language? }?,

attribute class { xsd:NMTOKENS }?,

attribute role { xsd:string }?,

attribute rel { xsd:string }?,

attribute rev { xsd:string }?,

attribute typeof { xsd:string }?,

attribute content { xsd:string }?,

attribute datatype { xsd:string }?,

attribute resource { xsd:string }?,

attribute about { xsd:string }?,

attribute property { xsd:string }?,

attribute xml:space { "default" | "preserve" }?,

attribute requiredFeatures { xsd:string }?,

attribute requiredExtensions { xsd:string }?,

attribute requiredFormats { xsd:string }?,

attribute requiredFonts { xsd:string }?,

attribute systemLanguage { xsd:string }?,

attribute transform { xsd:string | "none" }?,

attribute style { xsd:string }?,
attribute visibility {
  "visible" | "hidden" | "collapse" | "inherit" )?,
  # Added to SVG-1.2-RFC (for LibreOffice)
(desc 
  title
  path
  circle
  line
  ellipse
  polyline
  polygon
  solidColor
  textArea
  linearGradient
  radialGradient
  \text
  g
  defs
  use
  a)*
defs =
element defs {
  ((attribute fill-opacity { "inherit" | xsd:string }?,
  attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "none" | rfc-color }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke { rfc-color }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
    attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
    attribute stroke-miterlimit { "inherit" | xsd:string }?,
    attribute stroke-width { "inherit" | xsd:string }?,
    attribute color { rfc-color }?,
    attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
  & attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit" }?,
  & {attribute direction { "ltr" | "rtl" | "inherit" }?,
  Brownlee & IAB Expires April 18, 2016 [Page 45]
attribute unicode-bidi {
    "normal" | "embed" | "bidi-override" | "inherit"
}?
& (attribute solid-color { rfc-color }?,
attribute solid-opacity { "inherit" | xsd:string }?)
& (attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
}?,
attribute line-increment { "auto" | "inherit" | xsd:string }?)
& (attribute stop-color { rfc-color }?,
attribute stop-opacity { "inherit" | xsd:string }?)
& (attribute font-family { "inherit" | xsd:string }?,
attribute font-size { "inherit" | xsd:string }?,
attribute font-style {
    "normal" | "italic" | "oblique" | "inherit"
}?,
attribute font-variant { "normal" | "small-caps" | "inherit" }?,
attribute font-weight {
    "normal"
    | "bold"
    | "bolder"
    | "lighter"
    | "inherit"
}?,
attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
}?,
attribute text-align {
    "start" | "center" | "end" | "inherit"
}?)
(attribute id { xsd:NCName } |
attribute xml:id { xsd:NCName }?)?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute datatype { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
(desc
title
path)
rect
circle
line
ellipse
polyline
polygon
solidColor
textArea
linearGradient
radialGradient
text

g
defs
use
a)*
}
use =
element use {
(attribute fill-opacity { "inherit" | xsd:string }?,
attribute stroke-opacity { "inherit" | xsd:string }?)
& (attribute fill { "none" | rfc-color }?,
attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
attribute stroke { rfc-color }?,
attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
attribute stroke-dashoffset { "inherit" | xsd:string }?,
attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?)
& attribute vector-effect {
"none" | "non-scaling-stroke" | "inherit"
})
& {attribute direction { "ltr" | "rtl" | "inherit" }?,
attribute unicode-bidi { "normal" | "embed" | "bidi-override" | "inherit" }?)
& {attribute solid-color { rfc-color }?,
attribute solid-opacity { "inherit" | xsd:string }?)
& {attribute display-align { "auto" | "before" | "center" | "after" | "inherit" }?
attribute line-increment { "auto" | "inherit" | xsd:string }?,
& (attribute stop-color { rfc-color }?,
  attribute stop-opacity { "inherit" | xsd:string }?)?
& (attribute font-family { "inherit" | xsd:string }?,
  attribute font-size { "inherit" | xsd:string }?,
  attribute font-style { "normal" | "italic" | "oblique" | "inherit" }?,
  attribute font-variant { "normal" | "small-caps" | "inherit" }?,
  attribute font-weight { "normal" | "bold" | "bolder" | "lighter" | "inherit" }?),
attribute text-anchor { "start" | "middle" | "end" | "inherit" }?,
attribute text-align { "start" | "center" | "end" | "inherit" }?),
(attribute id { xsd:NCName } |
  attribute xml:id { xsd:NCName })?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute transform { xsd:string | "none" }?,
attribute ns1:show { "embed" }?,
attribute ns1:actuate { "onLoad" }?,
attribute ns1:type { "simple" }?,
attribute ns1:role { xsd:anyURI | xsd:string }?,
attribute ns1:show { "embed" }?,
attribute ns1:actuate { "onLoad" }?,
attribute ns1:type { "simple" }?,
attribute ns1:role { xsd:anyURI | xsd:string }?,

attribute ns1:arcrole { xsd:anyURI | xsd:string }?,
attribute ns1:title { text }?,
attribute ns1:href { xsd:anyURI | xsd:string }?,
attribute x { xsd:string }?,
attribute y { xsd:string }?,
(desc
 | title)*
}
a =
element a {
  (attribute id { xsd:NCName })
  | attribute xml:id { xsd:NCName })?,
  attribute xml:base { xsd:anyURI | xsd:string }?,
  attribute xml:lang { xsd:language? }?,
  attribute class { xsd:NMTOKENS }?,
  attribute role { xsd:string }?,
  attribute rel { xsd:string }?,
  attribute rev { xsd:string }?,
  attribute typeof { xsd:string }?,
  attribute content { xsd:string }?,
  attribute datatype { xsd:string }?,
  attribute resource { xsd:string }?,
  attribute about { xsd:string }?,
  attribute property { xsd:string }?,
  attribute xml:space { "default" | "preserve" }?,
  attribute requiredFeatures { xsd:string }?,
  attribute requiredExtensions { xsd:string }?,
  attribute requiredFormats { xsd:string }?,
  attribute requiredFonts { xsd:string }?,
  attribute systemLanguage { xsd:string }?,
  ((attribute fill-opacity { "inherit" | xsd:string }?),
  attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "none" | rfc-color }?),
  attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
  attribute stroke { rfc-color }?,
  attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
  attribute stroke-dashoffset { "inherit" | xsd:string }?,
  attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
  attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
  attribute stroke-miterlimit { "inherit" | xsd:string }?,
  attribute stroke-width { "inherit" | xsd:string }?,
  attribute color { rfc-color }?,
  attribute color-rendering { "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit" }?
& attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit"
}
& (attribute direction { "ltr" | "rtl" | "inherit" }?,
  attribute unicode-bidi {
    "normal" | "embed" | "bidi-override" | "inherit"
})?
& (attribute solid-color { rfc-color }?,
  attribute solid-opacity { "inherit" | xsd:string }?)
& (attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
}?,
  attribute line-increment { "auto" | "inherit" | xsd:string }?)
& (attribute stop-color { rfc-color }?,
  attribute stop-opacity { "inherit" | xsd:string }?)
& (attribute font-family { "inherit" | xsd:string }?,
  attribute font-size { "inherit" | xsd:string }?,
  attribute font-style { "normal" | "italic" | "oblique" | "inherit"
})?,
attribute font-variant { "normal" | "small-caps" | "inherit" }?,
attribute font-weight {
    "normal"
    | "bold"
    | "bolder"
    | "lighter"
    | "inherit"
},
attribute text-anchor {
    "start" | "middle" | "end" | "inherit"
}?,
attribute text-align {
    "start" | "center" | "end" | "inherit"
}?
),
attribute transform { xsd:string | "none" }?,
attribute ns1:show { "new" | "replace" }?,
attribute ns1:actuate { "onRequest" }?,
attribute ns1:type { "simple" }?,
attribute ns1:role { xsd:anyURI | xsd:string }?,
attribute ns1:arcrole { xsd:anyURI | xsd:string }?,
attribute ns1:title { text }?,
attribute ns1:href { xsd:anyURI | xsd:string }?,
attribute target {
    "_replace" | "_self" | "_parent" | "_top" | "_blank" | xsd:Name
},
desc
  title
Internet-Draft     SVG Drawings for RFCs: SVG 1.2 RFC       October 2015

path
rect
circle
line
ellipse
polyline
polygon
solidColor
textArea
linearGradient
radialGradient
\text
g
defs
use)*
)
stop =
element stop {
  ((attribute fill-opacity { "inherit" | xsd:string }?,
    attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "inherit" | xsd:string }?,
    attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
    attribute stroke { "inherit" | xsd:string }?,
    attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
    attribute stroke-dashoffset { "inherit" | xsd:string }?,
    attribute stroke-linecap {
      "butt" | "round" | "square" | "inherit"
    }?,
    attribute stroke-linejoin {
      "miter" | "round" | "bevel" | "inherit"
    }?,
    attribute stroke-miterlimit { "inherit" | xsd:string }?,
    attribute stroke-width { "inherit" | xsd:string }?,
    attribute color { rfc-color }?,
    attribute color-rendering {
      "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
    }?)
  & attribute vector-effect {
    "none" | "non-scaling-stroke" | "inherit"
  }?
  & {attribute direction { "ltr" | "rtl" | "inherit" }?,
    attribute unicode-bidi {
      "normal" | "embed" | "bidi-override" | "inherit"
    }?
  }
  & {attribute solid-color { "inherit" | xsd:string }?,
    attribute solid-opacity { "inherit" | xsd:string }?
  }
  & {attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
  }?
tspan =

(desc
 | title)*
)

tspan =
element tspan {
  ((attribute fill-opacity { "inherit" | xsd:string }?),
   attribute stroke-opacity { "inherit" | xsd:string }?)
   & (attribute fill { "none" | rfc-color }?),
   attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?),
}?
attribute stroke { rfc-color }?,
attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
attribute stroke-dashoffset { "inherit" | xsd:string }?,
attribute stroke-linecap {
  "butt" | "round" | "square" | "inherit"
}?,
attribute stroke-linejoin {
  "miter" | "round" | "bevel" | "inherit"
}?,
attribute stroke-miterlimit { "inherit" | xsd:string }?,
attribute stroke-width { "inherit" | xsd:string }?,
attribute color { rfc-color }?,
attribute color-rendering {
  "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
}?
& attribute vector-effect {
  "none" | "non-scaling-stroke" | "inherit"
}?
& {attribute direction { "ltr" | "rtl" | "inherit" }?,
    attribute unicode-bidi {
      "normal" | "embed" | "bidi-override" | "inherit"
    }?
}& {attribute solid-color { rfc-color }?,
    attribute solid-opacity { "inherit" | xsd:string }?
}& {attribute display-align {
    "auto" | "before" | "center" | "after" | "inherit"
}?,
    attribute line-increment { "auto" | "inherit" | xsd:string }?
}& {attribute stop-color { rfc-color }?,
    attribute stop-opacity { "inherit" | xsd:string }?
}& {attribute font-family { "inherit" | xsd:string }?,
    attribute font-size { "inherit" | xsd:string }?,
    attribute font-style {
      "normal" | "italic" | "oblique" | "inherit"
    }?,
    attribute font-variant { "normal" | "small-caps" | "inherit" }?,
    attribute font-weight {
      "normal"
      | "bold"
      | "bolder"
      | "lighter"
      | "inherit"
    }?,
    attribute text-anchor {
      "start" | "middle" | "end" | "inherit"
    }?,
    attribute text-align {
      "start" | "center" | "end" | "inherit"
attribute xml:id { xsd:NCName }?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute datatype { xsd:string }?,
attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute x { xsd:string }?, # For SVG-1.2-RFC
attribute y { xsd:string }?,
(tbreak
desc
tspan_2
text
| a_2)+
tspan_2 =
element tspan {
  ((attribute fill-opacity { "inherit" | xsd:string }?),
   attribute stroke-opacity { "inherit" | xsd:string }?)
& {attribute fill { "none" | rfc-color }?,
   attribute stroke { rfc-color }?,
   attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
   attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
   attribute stroke-dashoffset { "inherit" | xsd:string }?,
   attribute stroke-linecap { "butt" | "round" | "square" | "inherit" }?,
   attribute stroke-linejoin { "miter" | "round" | "bevel" | "inherit" }?,
   attribute stroke-miterlimit { "inherit" | xsd:string }?,
   attribute stroke-width { "inherit" | xsd:string }?,
   attribute color { rfc-color }?),

Brownlee & IAB Expires April 18, 2016 [Page 54]
attribute color-rendering {
   "auto" | "optimizeSpeed" | "optimizeQuality" | "inherit"
})
& attribute vector-effect {
   "none" | "non-scaling-stroke" | "inherit"
)?
& {attribute direction { "ltr" | "rtl" | "inherit" }?,
   attribute unicode-bidi {
      "normal" | "embed" | "bidi-override" | "inherit"
    )
& {attribute solid-color { rfc-color }?,
   attribute solid-opacity { "inherit" | xsd:string }?
& {attribute display-align {
      "auto" | "before" | "center" | "after" | "inherit"
    )?,
   attribute line-increment { "auto" | "inherit" | xsd:string }?
& {attribute stop-color { rfc-color }?,
   attribute stop-opacity { "inherit" | xsd:string }?
& {attribute font-family { "inherit" | xsd:string }?,
   attribute font-size { "inherit" | xsd:string }?,
   attribute font-style { "normal" | "italic" | "oblique" | "inherit"
    ),
   attribute font-variant { "normal" | "small-caps" | "inherit" }?,
   attribute font-weight {
      "normal"
      "bold"
      "bolder"
      "lighter"
      "inherit"
    )?,
   attribute text-anchor {
      "start" | "middle" | "end" | "inherit"
    )?,
   attribute text-align {
      "start" | "center" | "end" | "inherit"
    )
  },
(attribute id { xsd:NCName }
  | attribute xml:id { xsd:NCName })?,
attribute xml:base { xsd:anyURI | xsd:string }?,
attribute xml:lang { xsd:language? }?,
attribute class { xsd:NMTOKENS }?,
attribute role { xsd:string }?,
attribute rel { xsd:string }?,
attribute rev { xsd:string }?,
attribute typeof { xsd:string }?,
attribute content { xsd:string }?,
attribute datatype { xsd:string }?,

attribute resource { xsd:string }?,
attribute about { xsd:string }?,
attribute property { xsd:string }?,
attribute xml:space { "default" | "preserve" }?,
attribute requiredFeatures { xsd:string }?,
attribute requiredExtensions { xsd:string }?,
attribute requiredFormats { xsd:string }?,
attribute requiredFonts { xsd:string }?,
attribute systemLanguage { xsd:string }?,
attribute x { xsd:string }?,  # For SVG-1.2-RFC
attribute y { xsd:string }?,
(desc
  | title
  | tspan_2
  | text
  | a_2)+
} a_2 =
element a {
  (attribute id { xsd:NCName }
   | attribute xml:id { xsd:NCName })?,
  attribute xml:base { xsd:anyURI | xsd:string }?,
  attribute xml:lang { xsd:language? }?,
  attribute class { xsd:NMTOKENS }?,
  attribute role { xsd:string }?,
  attribute rel { xsd:string }?,
  attribute typeof { xsd:string }?,
  attribute content { xsd:string }?,
  attribute datatype { xsd:string }?,
  attribute resource { xsd:string }?,
  attribute about { xsd:string }?,
  attribute property { xsd:string }?,
  attribute xml:space { "default" | "preserve" }?,
  attribute requiredFeatures { xsd:string }?,
  attribute requiredExtensions { xsd:string }?,
  attribute requiredFormats { xsd:string }?,
  attribute requiredFonts { xsd:string }?,
  attribute systemLanguage { xsd:string }?,
  ((attribute fill-opacity { "inherit" | xsd:string }?,
    attribute stroke-opacity { "inherit" | xsd:string }?)
  & (attribute fill { "none" | rfc-color }?,
     attribute fill-rule { "inherit" | "nonzero" | "evenodd" }?,
     attribute stroke { rfc-color }?,
     attribute stroke-dasharray { "inherit" | "none" | xsd:string }?,
     attribute stroke-dashoffset { "inherit" | xsd:string }?,
     attribute stroke-linecap { "butt" | "round" | "square" | "inherit"}
attribute ns1:role { xsd:anyURI | xsd:string }?,
attribute ns1:arcrole { xsd:anyURI | xsd:string }?,
attribute ns1:title { text }?,
attribute ns1:href { xsd:anyURI | xsd:string }?,
attribute target {
        "_replace" | "_self" | "_parent" | "_top" | "_blank" | xsd:Name
    },
(desc
    | title
    | tspan_2
    | text)+
}
tbreak =
  element tbreak {
    (attribute id { xsd:NCName } |
        attribute xml:id { xsd:NCName })?,
    attribute xml:base { xsd:anyURI | xsd:string }?,
    attribute xml:lang { xsd:language? }?,
    attribute class { xsd:NMTOKENS }?,
    attribute role { xsd:string }?,
    attribute rel { xsd:string }?,
    attribute rev { xsd:string }?,
    attribute typeof { xsd:string }?,
    attribute content { xsd:string }?,
    attribute datatype { xsd:string }?,
    attribute resource { xsd:string }?,
    attribute about { xsd:string }?,
    attribute property { xsd:string }?,
    attribute xml:space { "default" | "preserve" }?,
    attribute requiredFeatures { xsd:string }?,
    attribute requiredExtensions { xsd:string }?,
    attribute requiredFormats { xsd:string }?,
    attribute requiredFonts { xsd:string }?,
    attribute systemLanguage { xsd:string }?
  }

--- End of SVG 1.2 RFC rnc schema

Authors’ Addresses

Nevil Brownlee
The University of Auckland

Email: n.brownlee@auckland.ac.nz
The Use of Non-ASCII Characters in RFCs

draft-flanagan-nonascii-06

Abstract

In order to support the internationalization of protocols and a more diverse Internet community, the RFC Series must evolve to allow for the use of non-ASCII characters in RFCs. While English remains the required language of the Series, the encoding of future RFCs will be in UTF-8, allowing for a broader range of characters than typically used in the English language. This document describes the RFC Editor requirements and guidance regarding the use of non-ASCII characters in RFCs.

This document updates RFC 7322.

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 May 21, 2016.

Copyright Notice

Copyright (c) 2015 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.
For much of the history of the RFC Series, the character encoding used for RFCs has been ASCII [ANSI.X3-4.1986]. This was a sensible choice at the time: the language of the Series has always been English, a language that primarily uses ASCII-encoded characters (ignoring for a moment words borrowed from more richly decorated alphabets); and, ASCII is the "lowest common denominator" for character encoding, making cross-platform viewing trivial.

There are limits to ASCII, however, that hinder its continued use as the exclusive character encoding for the Series. The increasing need for easily readable, internationalized content suggests it is time to allow non-ASCII characters in RFCs where necessary. To support this move away from ASCII, RFCs will switch to supporting UTF-8 as the character encoding.
default character encoding and allow support for a broad range of Unicode character support. [UnicodeCurrent] Note that the RFC Editor may reject any codepoint that does not render adequately in enough formats or on in enough rendering engines using the current tooling.

Given the continuing goal of maximum readability across platforms, the use of non-ASCII characters should be limited in a document to only where necessary within the text. This document describes the rules under which non-ASCII characters may be used in an RFC. These rules will be applied as the necessary changes are made to submission checking and editorial tools.

This document updates the RFC Style Guide [RFC7322].

The details described in this document are expected to change based on experience gained in implementing the RFC production center’s toolset. Revised documents will be published capturing those changes as the toolset is completed. Other implementers must not expect those changes to remain backwards-compatible with the details described this document.

2. Basic requirements

Two fundamental requirements inform the guidance and examples provided in this document. They are:

- Searches against RFC indexes and database tables need to return expected results and support appropriate Unicode string matching behaviors;
- RFCs must be able to display correctly across a wide range of readers and browsers. People whose system does not have the fonts needed to display a particular RFC need to be able to read the various publication formats and the XML correctly in order to understand and implement the information described in the document.

3. Rules for the use of non-ASCII characters

This section describes the guidelines for the use of non-ASCII characters in the header, body, and reference sections of an RFC. If the RFC Editor identifies areas where the use of non-ASCII characters negatively impacts the readability of the text, they will request alternate text.
The RFC Editor may, in cases of entire words represented in non-ASCII characters, ask for a set of reviewers to verify the meaning, spelling, characters, and grammar of the text.

3.1. General usage throughout a document

Where the use of non-ASCII characters is purely as part of an example and not otherwise required for correct protocol operation, escaping the non-ASCII character is not required. Note, however, that as the language of the RFC Series is English, the use of non-ASCII characters is based on the spelling of words commonly used in the English language following the guidance in the Merriam-Webster dictionary [MerrWeb].

Example of non-ASCII characters that do not require escaping [RFC4475]:

This particular response contains unreserved and non-ascii UTF-8 characters. This response is well formed. A parser must accept this message. Message Details : unreason SIP/2.0 200 = 2**3 * 5**2 &amp;#1085;&amp;#1086; &amp;#1089;&amp;#1090;&amp;#1086; &amp;#1076;&amp;#1077;&amp;#1074;&amp;#1103;&amp;#1085;&amp;#1086;&amp;#1089;&amp;#1090;&amp;#1086; &amp;#1076;&amp;#1077;&amp;#1074;&amp;#1103;&amp;#1090;&amp;#1100; - &amp;#1087;&amp;#1088;&amp;#1086;&amp;#1089;&amp;#1090;&amp;#1086;&amp;#1077; Via: SIP/2.0/UDP 192.0.2.198;branch=z9hG4bK1324923 Call-ID: unreason.1234ksdfak3j2erwedfsASdf CSeq: 35 INVITE From: sip:user@example.com;tag=11141343 To: sip:user@example.edu;tag=2229 Content-Length: 154 Content-Type: application/sdp

3.2. Authors, Contributors, and Acknowledgments

Person names may appear in several places within an RFC. In all cases, valid Unicode is required. For names that include non-ASCII characters, an author-provided, ASCII-only identifier is required to assist in search and indexing of the document.

Example for the header:
Example for the Acknowledgements:

OLD: The following people contributed significant text to early versions of this draft: Patrik Faltstrom, William Chan, and Fred Baker.

PROPOSED/NEW: The following people contributed significant text to early versions of this draft: Patrik Faeltstroem (Patrik Faltstrom), … (William Chan), and Fred Baker.

3.3. Company Names

Company names may appear in several places within an RFC. The rules for company names follow similar guidance to that of person names. Valid Unicode is required. For company names that include non-ASCII characters, an ASCII-only identifier is required to assist in search and indexing of the document.

3.4. Body of the document

When the mention of non-ASCII characters is required for correct protocol operation and understanding, the characters’ Unicode character name or code point MUST be included in the text.

o Non-ASCII characters will require identifying the Unicode code point.

o Use of the actual UTF-8 character (e.g., \u0096) is encouraged so that a reader can more easily see what the character is, if their device can render the text.

o The use of the Unicode character names like "INCREMENT" in addition to the use of Unicode code points is also encouraged. When used, Unicode character names should be in all capital letters.

Examples:
OLD [RFC7564]:

However, the problem is made more serious by introducing the full range of Unicode code points into protocol strings. For example, the characters U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 from the Cherokee block look similar to the ASCII characters "STPETER" as they might appear when presented using a "creative" font family.

NEW/ALLOWED:

However, the problem is made more serious by introducing the full range of Unicode code points into protocol strings. For example, the characters U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2 (#5082;#5026;#5045;#5036;#5026;#5036;#5074;) from the Cherokee block look similar to the ASCII characters "STPETER" as they might appear when presented using a "creative" font family.

ALSO ACCEPTABLE:

However, the problem is made more serious by introducing the full range of Unicode code points into protocol strings. For example, the characters "ᏚᎢᎵᎪᎢᏒ" (U+13DA U+13A2 U+13B5 U+13AC U+13A2 U+13AC U+13D2) from the Cherokee block look similar to the ASCII characters "STPETER" as they might appear when presented using a "creative" font family.

Example of proper identification of Unicode characters in an RFC:

Acceptable:

- Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character.

Preferred:

1. Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character ("\#916;").

2. Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character (INCREMENT).

3. Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character ("\#916", INCREMENT).

4. Temperature changes in the Temperature Control Protocol are indicated by the U+2206 character (INCREMENT, "\#916;").
5. Temperature changes in the Temperature Control Protocol are indicated by the "Delta" character "&amp;#916;" (U+2206).

6. Temperature changes in the Temperature Control Protocol are indicated by the character "&amp;#916;" (INCREMENT, U+2206).

Which option of (1), (2), (3), (4), (5), or (6) is preferred may depend on context and the specific character(s) in question. All are acceptable within an RFC. BCP 137, "ASCII Escaping of Unicode Character" describes the pros and cons of different options for identifying Unicode characters in an ASCII document BCP137 [RFC5137].

3.5. Tables

Tables follow the same rules for identifiers and characters as in "Section 3.4. Body of the document". If it is sensible (i.e., more understandable for a reader) for a given document to have two tables -- one including the identifiers and non-ASCII characters and a second with just the non-ASCII characters -- that will be allowed on a case-by-case basis.

Original text from "Preparation, Enforcement, and Comparison of Internationalized Strings Representing Usernames and Passwords" [RFC7613].

Table 3: A sample of legal passwords

<table>
<thead>
<tr>
<th>#</th>
<th>Password</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>&lt;correct horse battery staple&gt;</td>
<td>ASCII space is allowed</td>
</tr>
<tr>
<td>13</td>
<td>&lt;Correct Horse Battery Staple&gt;</td>
<td>Different from example 12</td>
</tr>
<tr>
<td>14</td>
<td>&lt;π&amp;#xDF;&amp;#xE5;&gt;</td>
<td>Non-ASCII letters are OK (e.g., GREEK SMALL LETTER PI, U+03C0)</td>
</tr>
<tr>
<td>15</td>
<td>&lt;Jack of &amp;#x2666;&amp;s&gt;</td>
<td>Symbols are OK (e.g., BLACK DIAMOND SUIT, U+2666)</td>
</tr>
<tr>
<td>16</td>
<td>&lt;foo&amp;x1680;bar&gt;</td>
<td>OGHAM SPACE MARK, U+1680, is mapped to U+0020 and thus the full string is mapped to &lt;foo bar&gt;</td>
</tr>
</tbody>
</table>

Preferred text:
Table 3: A sample of legal passwords

<table>
<thead>
<tr>
<th>#</th>
<th>Password</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>&lt;correct horse battery staple&gt;</td>
<td>ASCII space is allowed</td>
</tr>
<tr>
<td>13</td>
<td>&lt;Correct Horse Battery Staple&gt;</td>
<td>Different from example 12</td>
</tr>
<tr>
<td>14</td>
<td>&lt;πss๗&gt;</td>
<td>Non-ASCII letters are OK (e.g., GREEK SMALL LETTER PI, U+03C0; LATIN SMALL LETTER SHARP S, U+00DF; THAI DIGIT SEVEN, U+0E57)</td>
</tr>
<tr>
<td>15</td>
<td>&lt;Jack of ♦s&gt;</td>
<td>Symbols are OK (e.g., BLACK DIAMOND SUIT, U+2666)</td>
</tr>
<tr>
<td>16</td>
<td>&lt;foo bar&gt;</td>
<td>OGHAM SPACE MARK, U+1680, is mapped to U+0020 and thus the full string is mapped to &lt;foo bar&gt;</td>
</tr>
</tbody>
</table>

3.6. Code components

The RFC Editor encourages the use of the U+ notation except within a code component where you must follow the rules of the programming language in which you are writing the code.

3.7. Bibliographic text

The reference entry must be in English; whatever subfields are present must be available in ASCII-encoded characters. As long as good sense is used, the reference entry may also include non-ASCII characters at the author’s discretion and as provided by the author. The RFC Editor may request a review of the non-ASCII reference entry. This applies to both normative and informative references.

Example:
Internet-Draft              non-ASCII in RFCs              November 2015

[81x742] Internet-Draft              non-ASCII in RFCs              November 2015


Allowable addition to the above citation:

"Государственное комитето Российской Федерации по стандартизации и метрологии. "Информационная технология. шифрование сообщений. Подписи."

3.8. Keywords and Citation Tags

Keywords and citation tags must be ASCII only.

3.9. Address Information

The purpose of providing address information, either postal or e-mail, is to assist readers of an RFC to contact the author or authors. Authors may include the official postal address as recognized by their company or local postal service without additional non-ASCII character escapes. If the email address includes non-ASCII characters and is a valid email address at the time of publication, non-ASCII character escapes are not required.

4. Normalization Forms

Authors should not expect normalization forms to be preserved. If a particular normalization form is expected, note that in the text of the RFC.

5. XML Markup

As described above, use of non-ASCII characters in areas such as email, company name, addresses, and name is allowed. In order to make it easier for code to identify the appropriate ASCII alternatives, authors must include an "ascii" attribute to their XML markup when an ASCII alternative is required. See [I-D.hoffman-xml2rfc] for more detail on how to tag ASCII alternatives.

6. IANA Considerations

This document makes no request of IANA.

Note to RFC Editor: this section may be removed on publication as an RFC.
7. Internationalization Considerations

The ability to use non-ASCII characters in RFCs in a clear and consistent manner will improve the ability to describe internationalized protocols and will recognize the diversity of authors. However, the goal of readability will override the use of non-ASCII characters within the text.

8. Security Considerations

Valid Unicode that matches the expected text must be verified in order to preserve expected behavior and protocol information.

9. Change log - to be removed by the RFC Editor

9.1. -04 to -05

Keywords: expanded section to include citation tags.

Internationalization considerations: reiterated that the use of non-ASCII characters is not automatically guaranteed.

9.2. -04 to -05

Introduction: added statement regarding document subject to change.

Tables: added example.

Code: removed placeholder for example.

9.3. -02 to -04

Introduction and Abstract: change to be clearer about what/why non-ASCII characters are being allowed.

XML Markup: section added.

10. References

[ANSI.X3-4.1986]

[I-D.hoffman-xml2rfc]
Appendix A. Acknowledgements

With many thanks to the members of the IAB i18n program and the RFC Format Design Team.
Author’s Address

Heather Flanagan (editor)
RFC Editor

Email: rse@rfc-editor.org
URI: http://orcid.org/0000-0002-2647-2220
Abstract

In order to meet the evolving needs of the Internet community, the format for RFCs is changing from a plain-text, ASCII-only format to a canonical XML format that will in turn be rendered into several publication formats. This document defines the HTML format that will be rendered for an RFC or Internet-Draft.

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 February 25, 2016.

Copyright Notice

Copyright (c) 2015 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 ........................................... 4
2. Requirements for the HTML Format ....................... 5
  2.1. Requirements for Accessibility ........................ 6
3. HTML Version ............................................ 7
4. HTML Syntax ............................................. 7
5. Common Items ............................................ 8
  5.1. IDs .................................................. 8
  5.2. Pilcrows ............................................. 8
  5.3. ASCII Equivalents ................................... 8
  5.4. Syntactic Elements .................................. 9
6. Front Matter ............................................. 9
  6.1. DOCTYPE ............................................ 9
  6.2. Root Element ....................................... 10
  6.3. Head Element ....................................... 10
    6.3.1. Charset Declaration ............................ 10
    6.3.2. Document Title ................................ 10
    6.3.3. Document metadata ................................ 10
    6.3.4. Style ........................................... 11
    6.3.5. Links ........................................... 11
  6.4. Document Information ................................ 11
  6.5. Table of Contents ................................... 12
7. Main Body .............................................. 13
8. Back Matter ............................................. 13
  8.1. Index ................................................ 13
    8.1.1. Index Contents ................................ 13
    8.1.2. Index Letters ................................... 13
    8.1.3. Index Items .................................... 14
    8.1.4. Index Sub-items ................................ 14
  8.2. Authors’ Addresses ................................... 15
  8.3. Document Information ................................ 16
  8.4. XML Source .......................................... 17
9. Elements ............................................... 17
  9.1. <abstract> .......................................... 17
  9.2. <address> ............................................ 18
  9.3. <annotation> ........................................ 18
  9.4. <area> ............................................... 18
  9.5. <artwork> ............................................ 18
    9.5.1. Text Artwork .................................... 18
    9.5.2. SVG Artwork ..................................... 19
    9.5.3. Other Artwork ................................... 19
  9.6. <aside> .............................................. 20
  9.7. <author> ............................................. 20
    9.7.1. Authors of this document .......................... 20
9.7.2. Authors of references ........................................ 21
9.8. <back> ............................................................. 21
9.9. <bcp14> ............................................................. 21
9.10. <blockquote> ......................................................... 21
9.11. <boilerplate> ...................................................... 21
9.12. <br> ................................................................. 22
9.13. <city> ............................................................... 22
9.14. <code> ............................................................... 22
9.15. <country> ........................................................... 22
9.16. <crlf> ................................................................. 22
9.17. <date> ............................................................... 23
9.18. <dd> ................................................................. 23
9.19. <displayreference> .................................................. 23
9.20. <dl> ................................................................. 23
9.21. <dt> ................................................................. 23
9.22. <em> ................................................................. 23
9.23. <email> .............................................................. 23
9.24. <eref> .............................................................. 24
9.25. <figure> ............................................................ 24
9.26. <front> ............................................................. 24
9.27. <iref> .............................................................. 24
9.28. <keyword> .......................................................... 24
9.29. <li> ................................................................. 25
9.30. <link> ............................................................... 25
9.31. <middle> ............................................................ 25
9.32. <name> .............................................................. 25
9.33. <note> .............................................................. 25
9.34. <ol> ................................................................. 25
  9.34.1. Percent styles .................................................. 26
  9.34.2. Standard styles ................................................ 26
9.35. <organization> ...................................................... 26
9.36. <phone> ............................................................ 26
9.37. <postal> ........................................................... 26
9.38. <postalLine> ....................................................... 27
9.39. <refcontent> ........................................................ 27
9.40. <reference> ........................................................ 28
9.41. <referencegroup> .................................................... 28
9.42. <references> ......................................................... 28
9.43. <region> ............................................................ 29
9.44. <relref> ............................................................ 29
  9.44.1. displayFormat='of' ............................................. 29
  9.44.2. displayFormat='comma' ........................................ 30
  9.44.3. displayFormat='paren' ........................................ 30
  9.44.4. displayFormat='bare' ......................................... 31
9.45. <rfc> .............................................................. 31
9.46. <section> .......................................................... 31
9.47. <seriesInfo> ......................................................... 32
9.48. <sourcecode> ....................................................... 32
1. Introduction

As described in [I-D.flanagan-rfc-framework], the RFC Series is changing. One of those changes includes the RFC Editor publishing a non-canonical HTML version of RFCs.

This document describes the HTML format that will be used as one of the publication formats for the RFC Series. It defines a strict subset of HTML appropriate for RFC Series documents. The visual layout of the document will be defined through a cascading style sheet (CSS) [W3C.REC-CSS2-20110607]. The CSS will be included in the HTML file but will be described in a separate document.

The details (particularly any vocabularies) described in this document are expected to change based on experience gained in implementing the RFC production center’s toolset. Revised documents will be published capturing those changes as the toolset is completed. Other implementers must not expect those changes to remain backwards-compatible with the details described this document.
2. Requirements for the HTML Format

This section lists the design requirements used to create the HTML format described in this document. These requirements build on those found in [RFC6949]. Many of these requirements are naturally fulfilled by using the output of the prep tool [I-D.hoffman-rfcv3-preptool].

- The HTML has to render correctly on a list of browser versions that the RFC Editor will keep up to date outside of this document.
- The format will consist of a subset of HTML deemed to be widely implemented by common browsers at the time the specification is created, likely to continue to be widely-implemented, and unlikely to cause security issues. This will maximize the chances that future HTML renderers (such as new web browsers) will continue to produce readable text from the HTML format without the format needing to be changed frequently.
- These requirements are expected to change in the future to reflect the expectation that HTML rendering will be required for current versions of browsers and platforms, while ideally continuing to render correctly on recent versions of those browsers.
- The HTML documents from the RFC Editor or Internet-Drafts directory may be re-rendered from the canonical XML format in the future to ensure the ongoing readability of the documents. The intent is that any re-rendering would be due to exceptional circumstances rather than for minor annoyances.
- The HTML must display adequately in at least one text-based browser. Some consumers of the RFC series can only access the series on text-based terminals.
- The HTML document will be self-contained, without requiring external files for images, CSS, JavaScript, or the like. This will allow the HTML file to be moved over various non-HTTP transports (such as e-mail, FTP, and rsync) without breakage.
- Any use of JavaScript in the HTML document must not be required for comprehensive reading of the document, because some consumers of the RFC series routinely disable JavaScript for security purposes.
- The HTML document will allow easy local override of the default CSS formatting. This will allow users who have a different visual style that they prefer to make RFCs display with that style without having to alter the contents of the HTML document. This
might also be valuable for allowing people with specific accessibility needs to use a customized CSS.

- HTML tags in documents will rarely have attributes whose only purpose is to affect the rendered styling, and those will only be used if it would not be possible to specify that styling in CSS. No such attributes are known at this time.

- Both user-defined and auto-generated anchors must be supported and linkable, with user-defined anchors appearing in an "id" attribute. Auto-generated anchors will be generated for every heading, paragraph, and so on, not just those that do not have user-defined anchors. User-defined anchors may, and auto-generated anchors will, appear next to paragraphs, figures, tables, blockquotes, and section titles.

- All section, subsections, figures, and paragraphs should have stable numbered link anchors. Additionally, anchors expressed in the source XML should be exposed as anchors in the HTML output as well.

- The HTML must make it easy to separate sections along with all of their subsections into separate files. This will make creating EPUB documents easier in the future.

- The HTML produced for Internet-Drafts will differ from that produced by the RFC Editor due to differences in the output from the prep tool.

- The abstract must be marked up or tagged in a way that popular search engines will extract it as a summary.

2.1. Requirements for Accessibility

- Normative information must be easily accessible to the following consumers:
  - People with impaired vision, including those that use large fonts and those that use screen readers
  - People with difficulty distinguishing between colors
  - People who use devices with small screens, such as cell phones
  - Other groups to be determined later

- Specific instances where goals for accessibility are important in the design choices of the format have been called out in the text.
NOTE: designing for these consumers does not preclude the use of
features they cannot use, but does require that key semantic data
is not lost when read using the tools and settings that are
required by a given constituency.

3. HTML Version

The RFC Editor will periodically determine which version of the HTML
specification will be referenced for tools generating the format
defined in this document. The starting version will be that defined
in [W3C.REC-html5-20141028], commonly known as "HTML5". Although the
HTML specification mandates several of the syntax and structure rules
described in this document, they are called out here for emphasis.

4. HTML Syntax

The processor emitting HTML from the XML source will follow these
rules:

- The HTML output is encoded as UTF-8, as specified in [RFC3629].
- The document is valid HTML.
- Double quotes (U+0022 QUOTATION MARK: " ) are used to quote
  attribute values unless the HTML specification forbids quoting a
  particular attribute.
- Each logical line is terminated solely with a \n (U+000A: LINE
  FEED), otherwise known as "Unix-style" line endings.
- Other than \n (U+000A: LINE FEED), code points less than " "
  (U+0020: SPACE) (otherwise known as "control characters") are not
  used. Any character references that would generate these code
  points (such as & #0009;) may not be used. NOTE: this rule
  explicitly forbids \t (U+0009: CHARACTER TABULATION), \f (U+000C:
  FORM FEED), and \r (U+000D: CARRIAGE RETURN) from appearing in the
  output.
- Comments in the source XML, if any, will not be copied into the
  HTML.
- The HTML output will be pretty-printed, using whatever consistent
  rules deemed best in the HTML production tool.

NOTE: none of these rules affect the rendered output of the HTML, but
are intended to increase the chance that difference tools that
operate on the HTML output easier to write.
5. Common Items

This section lists items that are common across multiple parts of the HTML document.

5.1. IDs

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

5.2. Pilcrows

Each paragraph, artwork, or sourcecode segment outside of a <figure> or <table> element will be appended with a space and a "pilcrow" (U+00B6: PILCROW SIGN), otherwise known as a "paragraph sign". For the purposes of clarity, in this document pilcrows are rendered as "&para;". The pilcrow will linked to the "id" attribute on the XML entity to which it is associated.

The pilcrow will normally be invisible unless the element it is attached to is moused over. The pilcrow will be surrounded by a link that points to the element it is attached to.

Pilcrows are never included inside a <table> or <figure> elements, since the figure number or table number serve as adequate link targets.

Elements that might otherwise contain a pilcrow do not get marked with a pilcrow if they contain one or more child elements that are marked with a pilcrow. For example:

```html
<blockquote id="p-1.2-1">
  <p id="p-1.2-2">Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.</p>
  <a href="#p-1.2-2" class="pilcrow">&para;</a>
</blockquote>
```

5.3. ASCII Equivalents

Many elements in the v3 schema in [I-D.hoffman-xml2rfc] contain attributes for ASCII equivalents of the Unicode text contained in the element or the Unicode attribute value. These alternatives are...
5.4. Syntactic Elements

A few HTML elements are added to ensure particular syntax items can be styled appropriately using CSS.

Commas that are not in running text (e.g., that might have white-space added before them by the HTML indentation step) are wrapped by an HTML <span> tag of CSS class "comma".

Periods that are not in running text (e.g., that might have white-space added before them by the HTML indentation step) are wrapped by an HTML <span> tag of CSS class "fullStop".

Open and close parentheses that are not running text (e.g., that might have white-space added before them by the HTML indentation step) are wrapped by an HTML <span> tag of CSS class "openParen" or "closeParen" respectively.

6. Front Matter

The front matter of the HTML format contains processing information, metadata of various types, and styling information that applies to the document as a whole. This section describes HTML that is not necessarily a direct transform from the XML format. For more details on each of the tags that generate content in this section, see Section 9.

6.1. DOCTYPE

The DOCTYPE of the document is "html", which declares that the document is compliant with HTML5. The document will start with exactly this string:

<!DOCTYPE html>
6.2. Root Element

The root element of the document is <html>. This element includes a lang attribute, whose value is a [RFC5646] language tag describing the natural language of the document. The language tag to be included is "en". The class of the <html> element will be copied verbatim from the XML <rfc> element’s "mode" attribute, allowing CSS to style RFCs and Internet-Drafts differently from one another (if needed):

<html lang="en" class="RFC">

6.3. Head Element

The root <html> will contain a <head> element that contains the following elements, as needed.

6.3.1. Charset Declaration

In order to be correctly processed by browsers that load the HTML using a mechanism that does not provide a valid MIME content-type or charset (such as from a local file system using a "file:" URL), the HTML <head> element contains a <meta> element, with charset attribute with value "utf-8":

<meta charset="utf-8"/>

6.3.2. Document Title

The contents of the <title> element from the XML source will be placed inside an HTML <title> element in the header.

6.3.3. Document metadata

The following <meta> elements will be included:

- author – comma-separated <fullname>s of all of the <author>s from the XML source
- description – the <abstract> from the XML source
- generator – the name and version number of the software used to create the HTML
- keywords – comma-separated <keyword>s from the XML source

For example:
6.3.4. Style

The <head> element contains an embedded CSS style sheet in a <style> element. The styles in the style sheet are to be set consistently between documents by the RFC Editor, according to the best practices of the day.

To ensure consistent formatting, individual style attributes are not used in the main portion of the document except in highly exceptional circumstances; each use of such attributes will be individually justified.

Different readers of a specification will desire different formatting when reading the HTML versions of RFCs. To facilitate this, the <head> element also includes a <link> to a style sheet in the same directory as the HTML file, named "rfc-local.css". Any formatting in the linked style sheet will override the formatting in the included style sheet. For example:

```html
<style>
  body {}  
  ...
</style>
<link rel="stylesheet" type="text/css" href="rfc-local.css">
```

6.3.5. Links

Each <link> element from the XML source is copied into the HTML header.

6.4. Document Information

Information about the document as a whole will appear as the first child of the HTML <body> element, embedded in an HTML <dl> element with id="Identifiers". The defined terms in the definition list are "Workgroup:", "Series:", "Status:", "Published:", and "Authors:". For example:
<dl id="identifiers">
  <dt>Workgroup:</dt>
  <dd class="workgroup">rfc-interest</dd>
  <dt>Series:</dt>
  <dd class="series">Internet-Draft</dd>
  <dt>Status:</dt>
  <dd class="status">Informational</dd>
  <dt>Published:</dt>
  <dd><time datetime="2014-10-25" class="published">2014-10-25</time></dd>
  <dt>Authors:</dt>
  <dd class="authors">
    <div class="author">
      <span class="initial">J.</span> <span class="surname">Hildebrand</span>
      (Cisco Systems, Inc.)
    </div>
    <div class="author">
      <span class="initial">H.</span> <span class="surname">Flanagan</span>
      (RFC Editor)
    </div>
  </dd>
</dl>

6.5. Table of Contents

The table of contents will follow the boilerplate if the XML’s <rfc>
heading containing the text "Table of Contents" will be followed by a
<nav> element that contains a <ul> element for each depth of the
section hierarchy. Each section will be represented by a <li>
element containing links by the section number (from the "pn"
attribute) and by the name (from the "slugifiedName" attribute of the
<name> child element). Each <nav>, <ul>, and <li> element will have
the class "toc".

For example:

<h2 id="toc">Table of Contents</h2>
<nav class="toc">
  <ul class="toc">
    <li class="toc">
      <a href="s-1">1</a>. <a href="n-introduction">Introduction</a>
    </li>
    <li class="toc">
      <a href="s-1.1">1.1</a>. <a href="n-sub-intro">Sub Intro</a>
    </li>
    ...
  </ul>
</nav>
7. Main Body

The main body of the HTML document is processed according to the rules in Section 9.

8. Back Matter

The back matter of the HTML document includes an index (if generated), information about the authors, and further information about the document itself.

8.1. Index

The index will be produced at the end of the document (before the author information) if and only if the XML document’s <rfc> element has a indexInclude attribute with the value "true", and there is one or more <iref> elements in the document.

8.1.1. Index Contents

The index section will start with an <h2> heading containing the text "Index", followed by links to each of the lettered portions of the index. Links are not generated for letters that do not occur as the first letter of an index item.

For example:

```html
<h2>Index</h2>
<div class="index">
  <div class="indexIndex">
    <a href="#rfc.index.C">C</a>
    <a href="#rfc.index.P">P</a>
  </div>
  ...
</div>
```

8.1.2. Index Letters

The index index is followed by a <ul> tag that contains a <li> tag for each first letter represented in the index. This <li> tag has the class "indexChar", and contains an <a> tag with the id pointed to by the index index, as well as an href to itself. The <li> tag also includes a <ul> tag that will contain the index items.

For example:
8.1.3. Index Items

Each index item can have multiple <iref> elements to point to, all with the same item attribute. Each index item is represented by a <li> tag of class "indexItem" containing a <span> of class "irefItem" for the item text and one of class "irefRefs" for the generated references (if there is at least one reference to the item not having a subitem). Each generated reference contains an <a> tag containing the section number where the <iref> was found, with an "href" attribute pointing to the "irefid" attribute of the <iref> element from the XML document. If the primary attribute of the <iref> element has the value "true", the <a> element in the HTML document will have the class "indexPrimary". Commas may be used to separate the generated references, surrounded by a <span> tag with class "comma".

For example:

<li class="indexItem">
  <span class="irefItem">Bullets</span>
  <span class="irefRefs">
    <a class="indexPrimary" href="#s-Bullets-1">2</a>
    <span class="comma">,</span>
    <a href="#s-Bullets-2">2</a>
  </span>
</li>

8.1.4. Index Sub-items

If an index item has at least one subitem, the <li> of that item will contain a <ul>, with one <li> for each subitem, of class "indexSubItem". Each subitem is formatted similarly to items, except the class of the first <span> tag is "irefSubItem".

For example:


8.2. Authors’ Addresses

At the end of the document, author information will be included inside an HTML <section> element. The class names have been chosen to match the class names in [HCARD].

Note: The following example shows several ASCII equivalents that are the same as their nominal equivalents for clarity; normally the ASCII equivalents would not be included for these cases.

```html
<section id="author-addresses">
  <h2><a class="selfRef" href="#author-addresses">Authors’ Addresses</a></h2>
  <address class="vcard">
    <div class="nameRole">
      Joe Hildebrand <span class="ascii">Joe Hildebrand</span> (editor)
    </div>
    <div class="org">Cisco Systems, Inc.</div>
    <div class="adr">
      1 Main Street
      Suite 1
      Denver, CO
    </div>
  </address>
</section>
```
8.3. Document Information

A few bits of metadata about the document that are less important to most readers are included after the author information. The style sheet might de-emphasize their display, or hide them altogether. These are gathered together into a <div> of class "docInfo".

The finalized time is copied from the <rfc> element’s prepTime attribute. The rendered time is the time that this HTML was generated.

For example:
8.4.  XML Source

At the very end of the document, the XML source that was used to produce this document will be included within a comment. This comment may be preceded by another comment that describes the source.

Any instances of "--" in the XML will be modified to use U+002D characters: "--". Note that if the dashes in the original XML were in a comment start- (<!--) or end-delimiter (-->), the XML will not parse correctly without reversing this transformation.

<!-- XML SOURCE START (note: each instance of two '-' (U+002D: HYPHEN-MINUS) characters changed to "&amp;#x2d;&amp;#x2d;") -->
<!-- XML SOURCE START (note: each instance of two '-' (U+002D: HYPHEN-MINUS) characters changed to "&amp;#x2d;&amp;#x2d;") -->

9.  Elements

This section describes how each of the XML elements from [I-D.hoffman-xml2rfc] is rendered to HTML. Many of the descriptions have examples to clarify how elements will be rendered.

9.1.  <abstract>

The abstract is rendered similarly to a <section> with anchor="abstract" and <name>Abstract</name>, but without a section number.

<section id="abstract">
  <h2>a href="#abstract" class="selfRef">Abstract</a></h2>
  <p id="p-abstract-1">This document defines...
    <a href="#p-abstract-1" class="pilcrow">&para;</a></p>
</section>
9.2.  <address>

This element is used in Authors’ Addresses (Section 8.2).

9.3.  <annotation>

This element is rendered as a span of class "annotation" at the end
of a <reference> element, the span containing appropriately-
transformed elements from the children of the <annotation> tag. A
<span> of class "comma" is added before the annotation, containing
";".

<span class="comma">,</span>
<span class="annotation">
  You <span class="bcp14">MUST</span> read this annotation.
</span>

9.4.  <area>

Not currently rendered to HTML.

9.5.  <artwork>

Artwork can either consist of inline text or SVG. If the artwork is
not inside a <figure> element, a pilcrow (Section 5.2) is included.
Inside a <figure> element, the figure title serves the purpose of the
pilcrow. If the "align" attribute has the value "right", the CSS
class "alignRight" will be added. If the "align" attribute has the
value "center", the CSS class "alignCenter" will be added.

9.5.1.  Text Artwork

Text artwork is rendered inside an HTML <pre> element, which is
contained by a <div> element for consistency with SVG artwork. Note
that CDATA blocks do not work consistently in HTML, so all <, >, and
& must be escaped as &lt;, &gt;, and &amp;, respectively.

The <div> element will have CSS classes of "artwork" and "art-
" prepended to the value of the <artwork> element’s "type" attribute,
if it exists.
<figure id="f-1">
  <div class="artwork art-ascii-art" id="p-2-52">
    <pre>
      &lt;hello, world &gt;
      ----------------
      \  ^__^
      \ (oo)\_______
      (__)\       )\/
      | |----w |
      | |     |
    </pre>
    <figcaption>
      &lt;a href="#f-1">Figure 1.&lt;/a&gt;
      &lt;a class="selfRef" href="#n-it-figures" id="n-it-figures">It figures</a>
    </figcaption>
  </div>
</figure>

9.5.2. SVG Artwork

SVG artwork MUST be included inline. The SVG is wrapped in a <div> element with CSS classes "artwork" and "art-svg".

Note: the alt attribute of <artwork> is not currently used for SVG; instead, the <title> and <desc> tags are used in the SVG.

<svg width="100" height="100" xmlns="http://www.w3.org/2000/svg">
  <circle cx="50" cy="50" r="40"
        stroke="green" stroke-width="4" fill="yellow" />
</svg>

9.5.3. Other Artwork

Other artwork will have a src attribute whose value begins with "data:". Such artwork is rendered in an HTML image element.

Note: such images are not yet allowed by the RFC Series Editor, even though the format supports them.
9.6. <aside>

This element is rendered as an HTML <aside> element, with all child content appropriately transformed and a pilcrow (Section 5.2) added.

<aside id="p-1.2-6">A little more than kin, and less than kind.<para></a></aside>

9.7. <author>

9.7.1. Authors of this document

As seen in Authors’ Addresses (Section 8.2), at the end of the document, each document author is rendered into an HTML <address> element with the CSS class "vcard".

The HTML <address> element will contain an HTML <div> with CSS class "nameRole". That div will contain an HTML <span> element with CSS class "fn" containing the value of the "fullname" attribute of the <author> XML element, and an HTML <span> element with CSS class "role" containing the value of the "role" attribute of the <author> XML element (if there is a role). Parentheses will surround the <span class="role">, if it exists, each enclosed in a <span> with CSS class "openParen" or "closeParen" as appropriate.

<address class="vcard">
<div class="nameRole">
    <span class="fn">Joe Hildebrand</span>
    <span class="openParen">(</span>
    <span class="role">editor</span>
    <span class="closeParen">)</span>
</div>
... 

The <author> element from the <front> of the document is also rendered into the Document Information (Section 6.4), the HTML meta headers (Section 6.3.3), and in <references>. See each of those sections for details.
9.7.2. Authors of references

In the output generated from a reference element, author tags are rendered inside an HTML <span> element with CSS class "refAuthor".

9.8. <back>

This element does not add any direct output to HTML.

9.9. <bcp14>

This element marks up words like MUST and SHOULD with an HTML <span> element with the CSS class "bcp14".

You <span class="bcp14">MUST</span>/</span> be joking.

9.10. <blockquote>

This element renders as the similar HTML <blockquote> element. If there is a "cite" attribute, it is copied to the HTML cite attribute. If there is a "quoteFrom" attribute, it is placed inside a <cite> element at the end of the quote, with an <a> element surrounding it (if there is a "cite" attribute), linking to the "cite" URL.

If the blockquote does not contain another element that get a pilcrow (Section 5.2), a pilcrow is added.

Note that the "--" at the begining of the <cite> element should be a proper emdash, which is difficult to show in the current format of this format.

<blockquote id="p-1.2-1"
  cite="http://...">
  <p id="p-1.2-2">Four score and seven years ago our fathers brought forth on this continent, a new nation, conceived in Liberty, and dedicated to the proposition that all men are created equal.  
  <a href="#p-1.2-2" class="pilcrow">&para;</a>
  </p>
  <cite>-- <a href="http://...">Abraham Lincoln</a></cite>
</blockquote>

9.11. <boilerplate>

The IPR boilerplate for the document appears directly after the Abstract. The children of the input <boilerplate> element are treated similarly to sections.
This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

9.12. `<br>`

This element is directly rendered as its HTML counterpart.

9.13. `<city>`

This element is rendered as a `<span>` element with CSS class "locality".

<span class="locality">Guilford</span>

9.14. `<code>`

This element is rendered as a `<span>` element with CSS class "postal-code".

<span class="postal-code">GU16 7HF</span>

9.15. `<country>`

This element is rendered as a `<div>` element with CSS class "country-name".

<div class="country-name">England</div>

9.16. `<cref>`

This element is rendered as a `<span>` element with CSS class "cref". Any anchor is copied to the id attribute. If there is a source given, it is contained inside the cref span with another span of class "crefSource".

<span class="cref" id="crefAnchor">Just a brief comment about something that we need to remember later.</span>
<span class="crefSource">--life</span>
9.17. <date>

This element is rendered as the HTML <time> element. If the "year", "month", or "day" attribute is included on the XML element, an appropriate "datetime" element will be generated in HTML.

If this date is a child of the document’s <front> element, it gets the CSS class "published".

If this date is inside a <reference> element, it gets the CSS class "refDate".

<time datetime="2014-10" class="published">October 2014</time>

9.18. <dd>

This element is directly rendered as its HTML counterpart.

9.19. <displayreference>

This element is not rendered into HTML.

9.20. <dl>

This element is directly rendered as its HTML counterpart.

If the hanging attribute is "false", add the "dlParallel" class, else add the "dlHanging" class.

If the spacing attribute is "compact", add the "dlCompact" class.

9.21. <dt>

This element is directly rendered as its HTML counterpart.

9.22. <em>

This element is directly rendered as its HTML counterpart.

9.23. <email>

As shown in Section 8.2 this element is rendered as an HTML <div> containing the string "Email:" and an HTML <a> element, with "href" attribute set to the equivalent "mailto:" URI, CSS class of "email", and the contents set to the email address.

If the email contains an ascii attribute, a span of class ascii is also contained in the div.
9.24. <eref>

This element is rendered as HTML <a> element, with the "href" attribute set to the value of the "target" attribute, and the CSS class of "eref".

<a href="https://..." class="eref">the text</a>

9.25. <figure>

This element renders as the HTML <figure> element, containing the artwork or sourcecode indicated and an HTML <figcaption> element. The <figcaption> will contain an <a> element with CSS class "selfRef" around the figure number. It will also contain another <a> element with CSS class "selfRef" around the figure name, if a name was given.

<figure id="f-1">
  ...
  <figcaption>
    <a href="#f-1" class="selfRef">Figure 1.</a>
    <a href="#n-it-figures" class="selfRef">It figures</a>
  </figcaption>
</figure>

9.26. <front>

This element does not add any direct output to HTML.

9.27. <iref>

This element is rendered as an empty <> tag of class iref, with an id consisting of the <iref> element's irefid:

<span class="iref" id="s-Paragraphs-first-1"/>

9.28. <keyword>

Each of these elements renders its text into the <meta> keywords in the document’s header, separated by commas.

<meta name="keywords" content="html,css,rfc">

Hildebrand & Hoffman Expires February 25, 2016 [Page 24]
9.29. <li>
This element is rendered as its HTML counterpart, however if there is no contained element that had a pilcrow (Section 5.2) attached, a pilcrow is added.

<li id="p-2-7">Item <a href="#p-2-7" class="pilcrow">&para;</a></li>

9.30. <link>
This element is rendered as its HTML counterpart, in the HTML header.

9.31. <middle>
This element does not add any direct output to HTML.

9.32. <name>
This element is never rendered directly, but instead when considering its parent element, such as <section>.

9.33. <note>
This element is rendered similarly to a <section>, but without a section number, and with the CSS class of "note. If the "removeInRFC" attribute is set to "yes", the generated <div> will also include the CSS class "rfcEditorRemove".

<section id="s-note-1" class="note rfceditor-remove">
    <h2>
        <a href="#n-editorial-note" class="selfRef">Editorial Note</a>
    </h2>
    <p id="p-note-1-1">
        Discussion of this draft takes place...
        <a href="#p-note-1-1" class="pilcrow">&para;</a>
    </p>
</section>

9.34. <ol>
The output created from an <ol> element depends upon the style attribute.

If the spacing attribute has the value "compact", a CSS class of "olCompact" will be added.

The group attribute is not copied; the input XML should have start values added by a prep tool for all grouped <ol> elements.
9.34.1. Percent styles

If the style attribute includes the character "+", the output is a
<dl> tag with the class "olPercent". Each contained li is emitted as
a <dt>/<dd> pair, with the generated label in the <dt> and the
contents of the li in the <dd>.

<dl class="olPercent">
  <dt>Requirement xviii:</dt>
  <dd>Wheels on a big rig</dd>
</dl>

9.34.2. Standard styles

For all other styles, an <ol> tag is emitted, with any style
attribute turned into the equivalent HTML type attribute.

<ol class="compact" type="I" start="18">
  <li>Wheels on a big rig</li>
</ol>

9.35. <organization>

As shown in Section 8.2 this element is rendered as an HTML <div> tag
with CSS class "org".

<div class="org">Cisco Systems, Inc.</div>

9.36. <phone>

As shown in Section 8.2 this element is rendered as an HTML <div>
containing the string "Phone:" (wrapped in a span), an HTML <span>
with CSS class "tel" containing the phone number and an HTML <span>
with CSS class "type" containing the string "VOICE". Note, the
"type" span will be hidden by CSS styling.

<div>
  <span>Phone:</span>
  <span class="tel">+1-720-555-1212</span>
  <span class="type">VOICE</span>
</div>

9.37. <postal>

This element renders as an HTML <div> with CSS class "adr", unless it
contains a <postalline> child element, in which case it adds no HTML
markup.
When there is no `<postalLine>` child, the following child elements are rendered into the HTML:

- Each `<street>` is rendered
- A `<div>` that includes:
  - The rendering of all `<city>` elements
  - A comma (wrapped in a span of class "comma")
  - The rendering of all `<region>` elements
  - Whitespace
  - The rendering of all `<code>` elements
- The rendering of all `<country>` elements

```html
<div class="adr">
  <div class="street-address">1 Main Street</div>
  <div class="street-address">Suite 1</div>
  <div>
    <span class="city">Denver</span>,
    <span class="region">CO</span>
    <span class="postal-code">80212</span>
  </div>
  <div class="country-name">US</div>
</div>
```

9.38. `<postalLine>`

All of these elements in a give `<postal>` elements render as a single HTML `<pre>` with CSS class "label", with each `<postalLine>` separated by a newline. Note: this `<pre>` element is not enclosed in a `<div class="adr">`.

```html
<pre class="label">In care of:
  Computer Sciences Division</pre>
```

9.39. `<refcontent>`

This element renders as an HTML `<span>` with CSS class "refcontent".

```html
<span class="refContent">Self-published pamphlet</span>
```
9.40. <reference>

This element will render as a <dt> <dd> pair, with the defined term being the reference "anchor" attribute surrounded by square brackets, and the definition including the correct set of bibliographic information as specified by [RFC7322]. The <dt> element will have an "id" attribute of the reference anchor.

<dl class="reference">
  <dt id="RFC5646">[RFC5646]</dt>
  <dd>
    <span class="refAuthor">Phillips, A.</span> and <span class="refAuthor">M. Davis</span> <span class="refTitle">"Tags for Identifying Languages"</span>, <span class="comma"></span>
  </dd>
</dl>

9.41. <referencegroup>

A <referencegroup> is translated into a <span> of class "referenceGroup" which contains the references. <span> is used here to ensure that the reference lists remain as undisturbed as possible.

<dl class="reference">
  <dd>
  </dd>
</dl>

9.42. <references>

If there is at least one <references> element, a "References" section is added to the document, continuing with the next major section number after the last <section>.

Each references element will be added to that "References" section as if it were a section itself.
9.43. <region>
This element is rendered as a <span> element with CSS class "region".
</span class="region">Colorado</span>

9.44. <relref>
This element is rendered as one or more HTML <a> elements containing appropriate external links as their "href" attributes as well as (potentially) some connective text. All of the <a> elements generated will have class "relref". The contents of the <a> element(s) are determined by the values of the "derivedRemoteContent" and "displayFormat" attributes.

9.44.1. displayFormat='of'

The output is an <a> element with "href" attribute whose value is the value of the "derivedLink" attribute, and whose contents are the value of the "derivedRemoteContent" attribute. This is followed by the word "of" (surrounded by whitespace). This is followed by a second <a> element, surrounded by square brackets, whose href attribute is the value of the "target" attribute prepended with "#", and whose content is the value of the "target" attribute.

For example, if Section 2.3 of RFC 7878 has the title "Protocol Overview", with an input of:

See <relref section="2.3" target="RFC7878" displayFormat="of" derivedLink="http://www.rfc-editor.org/info/rfc7878#s-2.3" derivedContent="Section 2.3"/> for an overview.

The HTML generated will be:
See \[<a class="relref" href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a> of [RFC7878]</a> for an overview.

### 9.44.2. displayFormat='comma'

The output is an `<a>` element with "href" attribute whose value is the value of the "target" attribute prepended by "#", and whose content is the value of the "target" attribute; the entire element is wrapped in square brackets. This is followed by a comma (",") followed by whitespace. This is followed by another `<a>` element whose "href" attribute is the value of the "derivedLink" attribute and whose content is the value of the "derivedContent" attribute.

For example, if Section 2.3 of RFC 7878 has the title "Protocol Overview", for an input of:

```
See <relref section="2.3" target="RFC7878" displayFormat="comma" derivedLink="http://www.rfc-editor.org/info/rfc7878#s-2.3" derivedContent="Section 2.3"/> for an overview.
```

The HTML generated will be:

```
See [RFC7878], <a class="relref" href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a>, for an overview.
```

### 9.44.3. displayFormat='parens'

The output is an `<a>` element with "href" attribute whose value is the value of the "target" attribute prepended by "#", and whose content is the value of the "target" attribute; the entire element is wrapped in square brackets. This is followed by whitespace. This is followed by another `<a>` element whose "href" attribute is the value of the "derivedLink" attribute and whose content is the value of the "derivedContent" attribute; the entire element is wrapped in parentheses.

For example, if Section 2.3 of RFC 7878 has the title "Protocol Overview", for an input of:

```
See <relref section="2.3" target="RFC7878" displayFormat="parens" derivedLink="http://www.rfc-editor.org/info/rfc7878#s-2.3" derivedContent="Section 2.3"/> for an overview.
```

The HTML generated will be:
See [<a class="relref" href="#RFC7878">RFC7878</a>]<a href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a>) for an overview.

9.44.4. displayFormat='bare'

The output is an <a> element whose "href" attribute is the value of the "derivedLink" attribute and whose content is the value of the "derivedContent" attribute.

For this input:

See <relref section="2.3" target="RFC7878" displayFormat="bare" derivedLink="http://www.rfc-editor.org/info/rfc7878#s-2.3" derivedContent="Section 2.3"/> and ...

The HTML generated will be:

See <a class="relref" href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a> and ...

9.45. <rfc>

Various attributes of this element are represented in different parts of the HTML document.

9.46. <section>

This element is rendered as an HTML <section> element, containing an appropriate level HTML heading element (<h2>-<h6>). That heading element contains a <a> element around the part number (pn), if applicable (for instance, <abstract> does not get a section number). Another <a> element is included with the section’s name.

<section id="intro">
  <h2 id="s-1">
    <a href="#s-1" class="selfRef">1.</a>
  </h2>
  <a href="#intro" class="selfRef">Introduction</a>
  <p id="p-1-1">Paragraph <a href="#p-1-1" class="pilcrow">&para;</a></p>
</section>
9.47. <seriesInfo>

This element is rendered in an HTML <span> element with CSS name "seriesInfo".

<span class="seriesInfo">RFC 5646</span>

9.48. <sourcecode>

This element is rendered in an HTML <pre> with a CSS class of "sourcecode". Note that CDATA blocks do not work consistently in HTML, so all <, >, and & must be escaped as &lt;, &gt;, and &amp;, respectively. If the input XML has a "type" attribute, another CSS class of "lang-" and the type is added.

If the sourcecode is not inside a <figure> element, a pilcrow (Section 5.2) is included. Inside a <figure> element, the figure title serves the purpose of the pilcrow.

<pre class="sourcecode lang-c">
#include &lt;stdio.h&gt;

int main(void)
{
    printf("hello, world\n");
    return 0;
}
</pre>

9.49. <street>

This element renders as an HTML <div> with CSS class "street-address".

<div class="street-address">1899 Wynkoop St, Suite 600</div>

9.50. <strong>

This element is directly rendered as its HTML counterpart.

9.51. <sub>

This element is directly rendered as its HTML counterpart.
9.52.  <sup>
This element is directly rendered as its HTML counterpart.

9.53.  <svg>
This element is rendered as part of the <artwork> element.

9.54.  <t>
This element is rendered as an HTML <p> element. A pilcrow (Section 5.2) is included.

<p id="p-1-1">A paragraph.
   <a href="#p-1-1" class="pilcrow">&para;</a></p>

9.55.  <table>
This element is directly rendered as its HTML counterpart.

9.56.  <tbody>
This element is directly rendered as its HTML counterpart.

9.57.  <td>
This element is directly rendered as its HTML counterpart.

9.58.  <tfoot>
This element is directly rendered as its HTML counterpart.

9.59.  <th>
This element is directly rendered as its HTML counterpart.

9.60.  <thead>
This element is directly rendered as its HTML counterpart.

9.61.  <title>
The title of the document appears in an <title> element in the <head> element, as shown in Section 6.3.2.

The title also appears in an <h1> element, and follows directly after the Document Information. The <h1> element has an id attribute with value "title".
Inside a reference, the title is rendered as an HTML <span> tag with CSS class "refTitle". The text is surrounded by quotes inside the <span>.

<span class="refTitle">"Tags for Identifying Languages"</span>

9.62. <tr>

This element is directly rendered as its HTML counterpart.

9.63. <tt>

This element is directly rendered as its HTML counterpart.

9.64. <ul>

This element is directly rendered as its HTML counterpart. If the "spacing" attribute has the value "compact", a CSS class of "ulCompact" will be added. If the "empty" attribute has the value "true", as CSS class of "ulEmpty" will be added.

9.65. <uri>

As shown in Section 8.2 this element is rendered as an HTML <div> containing the string "URI:" and an HTML <a> element, with "href" attribute set to the linked URI, CSS class of "url" [sic], and the contents set to the linked URI.

    <div>URI:
      <a href="http://www.example.com" class="url">http://www.example.com</a>
    </div>

9.66. <workgroup>

This element does not add any direct output to HTML.

9.67. <xref>

This element is rendered as an HTML <a> element containing an appropriate local link as the "href" attribute. The value of the "href" attribute is taken from the "target" attribute, prepended by "." The <a> element generated will have class "xref". The contents of the <a> element are the value of the "derivedContent" attribute. If the "format" attribute has the value "default", and the "target"
attribute points to a <reference> or <referencegroup> element, then the generated <a> element is surrounded by square brackets in the output.

<a class="xref" href="#target">Table 2</a>

or

[<a class="xref" href="#RFC1234">RFC1234</a>]

10. IANA Considerations

This document contains no actions for IANA

11. Security Considerations

Since RFCs are sometimes exchanged outside the normal Web sandboxing mechanism (such as using the "rsync" program to a mirror site) then loaded from a local file, more care must be taken with the HTML than is ordinary on the web.

12. Acknowledgments

Heather Flanagan was an early co-author of this document and helped its formation. The authors gratefully acknowledge the contributions of: Patrick Linskey, and the members of the RFC Format Design Team (Nevil Brownlee, Sandy Ginoza, Tony Hansen, Ted Lemon, Julian Reschke, Adam Roach, Alice Russo, Robert Sparks, Dave Thaler).

13. References

13.1. Normative References


[W3C.REC-CSS2-20110607]

[I-D.flanagan-rfc-framework]

13.2. Informative References


[I-D.hoffman-xml2rfc]

[I-D.hoffman-rfcv3-preptool]

Authors' Addresses

Joe Hildebrand (editor)
Cisco Systems, Inc.

Email: jhildebr@cisco.com
Paul Hoffman
ICANN

Email: paul.hoffman@icann.org
Abstract

This document defines the "XML2RFC" version 3 vocabulary; an XML-based language used for writing RFCs and Internet-Drafts. It is heavily derived from the version 2 vocabulary that is also under discussion. This document obsoletes the v2 grammar described in RFC 2629 and its expected followup, draft-iab-xml2rfc.

Editorial Note (To be removed by RFC Editor)

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

Status of This Memo

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 March 7, 2016.

Copyright Notice

Copyright (c) 2015 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
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 .... 8
       1.2.4. Additional Changes from v2 .................. 9
   1.3. Syntax Notation .................................... 10
2. Elements ................................................. 10
   2.1. <abstract> ........................................ 11
   2.2. <address> .......................................... 11
   2.3. <annotation> ....................................... 12
   2.4. <area> ............................................. 12
   2.5. <artwork> ......................................... 13
   2.6. <aside> ............................................ 16
   2.7. <author> ........................................... 17
   2.8. <back> ............................................. 18
   2.9. <bcp14> ............................................ 19
   2.10. <blockquote> ....................................... 19
   2.11. <boilerplate> .................................... 21
   2.12. <br> ............................................. 21
   2.13. <city> ............................................ 21
   2.14. <code> ........................................... 21
   2.15. <country> ........................................ 22
   2.16. <cref> ............................................ 22
   2.17. <date> ............................................ 23
   2.18. <dd> ............................................... 24
   2.19. <displayreference> ................................ 24
   2.20. <dl> ............................................... 26
   2.21. <dt> ............................................... 26
   2.22. <em> ............................................... 27
   2.23. <email> .......................................... 28
   2.24. <eref> ............................................ 29
   2.25. <figure> .......................................... 29
   2.26. <front> ........................................... 30
   2.27. <iref> ............................................ 30
   2.28. <keyword> ........................................ 34
   2.29. <li> ............................................... 34
2.30. <link> ....................................................... 36
2.31. <middle> .................................................... 37
2.32. <name> ....................................................... 37
2.33. <note> ......................................................... 37
2.34. <ol> .................................................................. 38
2.35. <organization> .................................................. 40
2.36. <phone> ........................................................... 41
2.37. <postal> ........................................................... 41
2.38. <postalLine> .................................................... 42
2.39. <refcontent> ..................................................... 42
2.40. <reference> ....................................................... 43
2.41. <referencegroup> ............................................... 44
2.42. <references> ...................................................... 44
2.43. <region> ............................................................ 45
2.44. <relref> ............................................................ 45
2.45. <rfc> ................................................................. 49
2.46. <section> .......................................................... 53
2.47. <seriesInfo> ...................................................... 55
2.48. <sourcecode> ..................................................... 57
2.49. <street> ............................................................ 59
2.50. <strong> ............................................................. 59
2.51. <sub> ................................................................. 60
2.52. <sup> ................................................................. 61
2.53. <t> ................................................................. 62
2.54. <table> ............................................................. 64
2.55. <tbody> ............................................................. 64
2.56. <td> ................................................................. 65
2.57. <tfoot> ............................................................ 67
2.58. <th> ................................................................. 67
2.59. <thead> ............................................................ 69
2.60. <title> ............................................................. 69
2.61. <tr> ................................................................. 70
2.62. <tt> ................................................................. 70
2.63. <ul> ................................................................. 71
2.64. <uri> ............................................................... 72
2.65. <workgroup> .................................................... 72
2.66. <xref> ............................................................. 72
2.67. <svg> (in namespace http://www.w3.org/2000/svg) .... 75
3.  Elements from v2 That Have Been Deprecated ............... 75
3.1. <c> ................................................................. 75
3.2. <facsimile> ....................................................... 76
3.3. <format> ........................................................... 76
3.4. <list> ............................................................... 76
3.5. <postamble> ....................................................... 77
3.6. <preamble> ........................................................ 78
3.7. <spanx> ........................................................... 78
3.8. <texttable> ........................................................ 79
3.9. <ttcol> ............................................................ 80
1. Introduction

This document describes version 3 ("v3") of the "XML2RFC" vocabulary; an XML-based language ('Extensible Markup Language', [XML]) used for writing RFCs ([RFC7322]) and Internet-Drafts ([IDGUIDE]).

This document obsoletes the version 2 vocabulary ("v2") [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.

The v3 format will be used as part of the new RFC series described in [RFC6949]. The new format will be handled by one or more new tools for preparing the XML and converting it to other representations. Features of the expected tools are described in Appendix B. That section defines some terms used throughout this document, such as "prep tool" and "formatter".

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

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

The details (particularly any vocabularies) described in this document are expected to change based on experience gained in implementing the RFC production center’s toolset. Revised documents will be published capturing those changes as the toolset is completed. Other implementers must not expect those changes to remain backwards-compatible with the details described in this document.

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

1.1. Design Criteria for the Changes in v3

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

- The intention is that starting and editing a v3 document will be easier than for a v2 document.
There will be good v2-to-v3 conversion tools for when an author wants to change versions.

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

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

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

1.2. Differences from v2 to v3

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

1.2.1. New Elements in v3

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

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

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

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

- Add <table>, <thead>, <tbody>, <tfoot>, <tr>, <td>, and <th> to give table functionality like that in HTML.

- Add <boilerplate> to hold the automatically-generated boilerplate text.
o Add <blockquote> to indicate a quotation as in a paragraph-like format.

o Add <name> to sections, notes, figures, and texttables to allow character formatting (fixed-width font) in their titles, and to allow references in the names.

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.

o Add <referencegroup> to allow referencing multi-RFC documents such as STDs and BCPs.

o Add <relref> to allow referencing specific sections or anchors in references.

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

o Add <br> to allow line breaks (but not blank lines) in the generated output for table cells.

o Add <svg> to allow easy inclusion of SVG drawings in <artwork>.

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. Add "prepTime" to indicate the time that the XML went through a preparation step. Add "version" to indicate the version of XML2RFC vocabulary used in the document. Add "scripts" to indicate which scripts are needed to render the document. Add "expiresDate" when an Internet Draft expires.

o Add "ascii" attributes to <email>, <organization>, <street>, <city>, <region>, <country>, and <code>. Also add "asciiFullname", "asciiInitials", and "asciiSurname" to <author>. 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 "anchor" attributes to many block elements to allow them to be linked with <relref> and <xref>.
1.2.3. Elements and Attributes Deprecated from v2

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

- Deprecate <list> in favor of <dl>, <ul>, and <ol>.
- Deprecate <spanx>; replace it with <strong>, <em>, and <tt>.
- Deprecate <vspace> because the major use for it, creating pseudo-paragraph-breaks in lists, is now handled properly.
- Deprecate <texttable>, <ttcol>, and <co>; replace them with the new table elements (<table> and the elements that can be contained within it).
- Deprecate <facsimile> because it is rarely used and is not actually useful; <email> is a much more useful way to get in touch with authors.
- Deprecate <format> because it is not useful and has caused surprise for authors in the past. If the goal is to provide a single URI (Uniform Resource Identifier) for a reference, use the "target" attribute on <reference> instead.
o Deprecate `<preamble>` and `<postamble>` in favor of simply using `<t>` before or after the figure. This also deprecates the "align" attribute in `<figure>`.

o Deprecate the "title" attribute in `<section>`, `<note>`, `<figure>`, `<references>`, and `<texttable>` in favor of the new `<name>`.

o Deprecate the "alt", and "src" 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 in both `<artwork>` and `<figure>` because they are not needed for the new output formats.

o Deprecate the "pageno" attribute in `<xref>` because it was unused in v2. Deprecate the "none" values for the "format" attribute in `<xref>` because it makes no sense semantically.

1.2.4. Additional Changes from v2

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

o Allow `<artwork>` and `<sourcecode>` to be used on their own in `<section>` (no longer confine them to a figure).

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

o Allow `<strong>`, `<em>`, `<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 `<section>`, restricted the names of the anchors that can be used on some types of sections.
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]), additionally constrained to US-ASCII characters ([USASCII]). Thus, the character repertoire consists of "A-Z", "a-z", "0-9", ",", ",", ",", ",", and ":", where "0-9", ",", and "," are disallowed as start character. Anchors are described in more detail in Appendix B.2.

Tools interpreting the XML described here will collapse horizontal whitespace and linebreaks to a single whitespace (except inside <artwork> and <sourcecode>), and will trim leading and trailing whitespace.

Some of the elements have attributes that are not described in this section because those elements are specific to the prep tool. People writing tools to process this format should read all of the appendices for a complete description of these attributes.

Every element in the v3 vocabulary can have an "xml:lang" attribute, an "xml:base" attribute, or both. The xml:lang attribute specifies the language used in the element. This is sometimes useful for renderers which display different fonts for ideographic characters.
used in China and Japan. The xml:base attribute is sometimes added to an XML file when doing XML-to-XML conversion where the base file has XInclude attributes (see Appendix B.1).

2.1. <abstract>

Contains the abstract of the document. See [RFC7322] for more information on restrictions for the abstract.

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

Content model:

In any order, but at least one of:

- o <dl> elements (Section 2.20)
- o <ol> elements (Section 2.34)
- o <t> elements (Section 2.53)
- o <ul> elements (Section 2.63)

2.1.1. ‘anchor’ attribute

Document-wide unique identifier for the abstract.

2.2. <address>

Provides address information for the author.

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

Content model:

In this order:

1. One optional <postal> element (Section 2.37)
2. One optional <phone> element (Section 2.36)
3. One optional <facsimile> element (Section 3.2)
4. One optional <email> element (Section 2.23)
5. One optional <uri> element (Section 2.64)
2.3. <annotation>

Provides additional prose augmenting a bibliographical reference. This text is intended to be shown after the rest of the generated reference text.

This element appears as a child element of: <reference> (Section 2.40).

Content model:

In any order:
  o Text
  o <bcp14> elements (Section 2.9)
  o <cref> elements (Section 2.16)
  o <em> elements (Section 2.22)
  o <eref> elements (Section 2.24)
  o <iref> elements (Section 2.27)
  o <relref> elements (Section 2.44)
  o <spanx> elements (Section 3.7)
  o <strong> elements (Section 2.50)
  o <sub> elements (Section 2.51)
  o <sup> elements (Section 2.52)
  o <tt> elements (Section 2.62)
  o <xref> elements (Section 2.66)

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 full name or the abbreviation of one of the IETF areas as listed on <http://www.ietf.org/iesg/area.html>. The list will be kept by the RFC Editor.
This element appears as a child element of: <front> (Section 2.26).

Content model: only text content.

2.5. <artwork>

This element allows the inclusion of "artwork" into the document. <artwork> provides full control of horizontal whitespace and line breaks, and thus is used for a variety of things, such as diagrams ("line art") and protocol unit diagrams.

Alternatively, the "src" attribute allows referencing an external graphics file, such as a vector drawing in SVG or a bitmap graphic file, using a URI. In this case, the textual content acts as fallback for output representations that do not support graphics, and thus ought to contain either a "line art" variant of the graphics, or otherwise prose that describes the included image in sufficient detail.

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

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

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

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

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

- As a data: URI, such as: <artwork type="svg" src="data:image/svg+xml,%3Csvg xmlns="http://www.w3.org/2000/svg"%3E%3C%3...">

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

- As a local file, such as: <artwork type="svg" src="diagram12.svg">

The use of SVG in Internet Drafts and RFCs is covered in much more detail in [SVGforRFCs].

The above methods for inclusion of SVG art can also be used for
including text artwork, but using a data: URI is probably confusing for text artwork.

Formatters that do pagination should attempt to keep artwork on a single page. This is to prevent artwork that is split across pages from looking like two separate pieces of artwork.

This element appears as a child element of: <aside> (Section 2.6), <blockquote> (Section 2.10), <dd> (Section 2.18), <figure> (Section 2.25), <li> (Section 2.29), <section> (Section 2.46), <td> (Section 2.56), and <th> (Section 2.58).

Content model:

Either:

Text

Or:

<svg> elements (Section 2.67)

2.5.1. ‘align’ attribute

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

Allowed values:

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

2.5.2. ‘alt’ attribute

Alternative text description of the artwork (which is more than just a summary or caption). When the art comes from the "src" attribute, and the format of that artwork supports alternate text, the alternative text comes from the text of the artwork itself, not from this attribute. The contents of this attribute are important to readers who are visually impaired, as well as those reading on devices that cannot show the artwork well, or at all.
2.5.3. ‘anchor’ attribute

Document-wide unique identifier for this artwork.

2.5.4. ‘height’ attribute

Deprecated.

2.5.5. ‘name’ attribute

A filename suitable for the contents (such as for extraction to a local file). This attribute can be helpful for other kinds of tools (such as automated syntax checkers which work by extracting the artwork). Note that the "name" attribute does not need to be unique for artwork elements in a document. If multiple artwork elements have the same name attribute, a processing tool might assume that the elements are all fragments of a single file, and the tool can collect those fragments for later processing. See Section 5 for a discussion of possible problems with the value of this attribute.

2.5.6. ‘src’ attribute

The URI reference of a graphics file ([RFC3986]), or the name of a file on the local disk. This can be a "data" URI ([RFC2397]) that contains the contents of the graphics file. Note that the inclusion of art with the "src" attribute depends on the capabilities of the processing tool reading the XML document. Tools need to be able to handle the file: URI, and should be able to handle http: and https: URIs as well. The prep tool will be able to handle reading the "src" attribute.

If no URI scheme is given in the attribute, the attribute is considered to be a local file name. Processing tools must be careful to not accept dangerous values for the filename, particularly those that contain absolute references outside the current directory.

In some cases, the prep tool may remove the "src" attribute after processing its value. See [PREPTOOL] for a description of this.

It is an error to have both a "src" attribute and content in the <artwork> element.

2.5.7. ‘type’ attribute

Specifies the type of the artwork. The value of this attribute is free text with certain values designated as preferred.

The preferred values for <artwork> types are:
o ascii-art
o binary-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.

2.5.8. ‘width’ attribute

Deprecated.

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.

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

Content model:

In any order:

o <artwork> elements (Section 2.5)

o <dl> elements (Section 2.20)

o <figure> elements (Section 2.25)

o <iref> elements (Section 2.27)

o <list> elements (Section 3.4)

o <ol> elements (Section 2.34)

o <t> elements (Section 2.53)
2.6.1. 'anchor' attribute

Document-wide unique identifier for this aside.

2.7. <author>

Provides information about a document’s author. This is used both for the document itself (at the beginning of the document) and for referenced documents.

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 [RFC7322]).

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

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

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

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

Content model:

In this order:

1. One optional <organization> element (Section 2.35)
2. One optional <address> element (Section 2.2)

2.7.1. 'asciiFullname' attribute

The ASCII equivalent of the author’s full name.

2.7.2. 'asciiInitials' attribute

The ASCII equivalent of the author’s initials.
2.7.3. 'asciiSurname' attribute

The ASCII equivalent of the author’s surname.

2.7.4. 'fullname' attribute

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

2.7.5. 'initials' attribute

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

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

2.7.6. 'role' attribute

Specifies the role the author had in creating the document.

Allowed values:

- "editor"

2.7.7. 'surname' attribute

The author’s surname (used on the front page and in references).

2.8. <back>

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

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

Content model:

In this order:

1. Optional <displayreference> elements (Section 2.19)
2. Optional <references> elements (Section 2.42)
3. Optional <section> elements (Section 2.46)
2.9.  `<bcp14>`

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

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

This element appears as a child element of: `<annotation>` (Section 2.3), `<blockquote>` (Section 2.10), `<dd>` (Section 2.18), `<dt>` (Section 2.21), `<em>` (Section 2.22), `<li>` (Section 2.29), `<preamble>` (Section 3.6), `<refcontent>` (Section 2.39), `<strong>` (Section 2.50), `<sub>` (Section 2.51), `<sup>` (Section 2.52), `<t>` (Section 2.53), `<td>` (Section 2.56), `<th>` (Section 2.58), and `<tt>` (Section 2.62).

Content model: only text content.

2.10.  `<blockquote>`

Specifies a block of text is a quotation.

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

Content model:

Either:

- In any order, but at least one of:

  * `<artwork>` elements (Section 2.5)
  * `<dl>` elements (Section 2.20)
  * `<figure>` elements (Section 2.25)
  * `<ol>` elements (Section 2.34)
  * `<sourcecode>` elements (Section 2.48)
  * `<t>` elements (Section 2.53)
  * `<ul>` elements (Section 2.63)
Or:

In any order, but at least one of:

* Text
* <bcp14> elements (Section 2.9)
* <cref> elements (Section 2.16)
* <em> elements (Section 2.22)
* <eref> elements (Section 2.24)
* <iref> elements (Section 2.27)
* <relref> elements (Section 2.44)
* <strong> elements (Section 2.50)
* <sub> elements (Section 2.51)
* <sup> elements (Section 2.52)
* <tt> elements (Section 2.62)
* <xref> elements (Section 2.66)

2.10.1. ‘anchor’ attribute

Document-wide unique identifier for this quotation.

2.10.2. ‘cite’ attribute

The source of the citation. This must be a URI. If the quotedFrom attribute is given, this URI will be used by processing tools as the link for the text of that attribute.

2.10.3. ‘quotedFrom’ attribute

Name of person or document the text in this element is quoted from. A formatter should render this as visible text at the end of the quotation.
2.11.  <boilerplate>

Holds the boilerplate text for the document. This section is filled in by the prep tool.

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

Content model:

One or more <section> elements (Section 2.46)

2.12.  <br>

Indicates that a line break should be inserted in the generated output by a formatting tool. Multiple successive instances of this element do not cause blank lines to appear in the output, and is thus not useful.

This element appears as a child element of: <td> (Section 2.56), and <th> (Section 2.58).

Content model: this element does not have any contents.

2.13.  <city>

Gives the city name in a postal address.

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

Content model: only text content.

2.13.1.  ‘ascii’ attribute

The ASCII equivalent of the city name.

2.14.  <code>

Gives the postal region code.

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

Content model: only text content.

2.14.1.  ‘ascii’ attribute

The ASCII equivalent of the postal code.
2.15. <country>

Gives the country name or code in a postal address.

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

Content model: only text content.

2.15.1. ‘ascii’ attribute

The ASCII equivalent of the country name.

2.16. <cref>

Represents a comment.

Comments can be used in a document while it is work-in-progress. They might appear either inline and visually highlighted, at the end of the document, or not at all, depending on the formatting tool.

This element appears as a child element of: <annotation> (Section 2.3), <blockquote> (Section 2.10), <c> (Section 3.1), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <li> (Section 2.29), <name> (Section 2.32), <postamble> (Section 3.5), <preamble> (Section 3.6), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <td> (Section 2.56), <th> (Section 2.58), <tt> (Section 2.62), and <ttcol> (Section 3.9).

Content model:

In any order:

- Text
- <em> elements (Section 2.22)
- <eref> elements (Section 2.24)
- <relref> elements (Section 2.44)
- <strong> elements (Section 2.50)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)
2.16.1. ‘anchor’ attribute

Document-wide unique identifier for this comment.

2.16.2. ‘display’ attribute

Suggests whether or not the the comment should be displayed by formatting tools. This might be set to "false" if you want to keep a comment in a document after the contents of the comment have already been dealt with.

Allowed values:

- "true" (default)
- "false"

2.16.3. ‘source’ attribute

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

2.17. <date>

Provides information about the publication date.

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

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

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

When the prep tool is used to create Internet Drafts, it will reject a submitted Internet Draft that has a <date> element in the
boilerplate for itself that is anything other than today. That is, the tool will not allow a submitter to specify a date other than the day of submission. To avoid this problem, authors might simply not include a <date> element in the boilerplate.

In the case of bibliographic references, the date information can have prose text for the month or year. For example, vague dates (year="ca. 2000"), date ranges (year="2012-2013"), non-specific months (month="Second quarter") and so on, are allowed.

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

Content model: this element does not have any contents.

2.17.1. ‘day’ attribute

In the "boilerplate" case: the day of publication; this is a number. Otherwise: an indication of the publication day, with the format not being restricted.

2.17.2. ‘month’ attribute

In the "boilerplate" case: the month of publication; this is the English name of the month. Otherwise: an indication of the publication month, with the format not being restricted.

2.17.3. ‘year’ attribute

In the "boilerplate" case: the year of publication; this is a number (usually four-digit). Otherwise: an indication of the publication year, with the format not being restricted.

2.18. <dd>

The definition part of an entry in a definition list.

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

Content model:

Either:

In any order, but at least one of:

* <artwork> elements (Section 2.5)
* <dl> elements (Section 2.20)
* <figure> elements (Section 2.25)
* <ol> elements (Section 2.34)
* <sourcecode> elements (Section 2.48)
* <t> elements (Section 2.53)
* <ul> elements (Section 2.63)

Or:

In any order, but at least one of:

* Text
* <bcp14> elements (Section 2.9)
* <cref> elements (Section 2.16)
* <em> elements (Section 2.22)
* <eref> elements (Section 2.24)
* <iref> elements (Section 2.27)
* <relref> elements (Section 2.44)
* <strong> elements (Section 2.50)
* <sub> elements (Section 2.51)
* <sup> elements (Section 2.52)
* <tt> elements (Section 2.62)
* <xref> elements (Section 2.66)

2.18.1. ‘anchor’ attribute

Document-wide unique identifier for this definition.
2.19.  <displayreference>

This element gives a mapping between the anchor of a reference and a
name that will be displayed instead. This allows authors to display
more 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 target="RFC6949" to="RFC-dev"/>

If a reference section is sorted, this element changes the sort
order.

This element appears as a child element of: <back> (Section 2.8).

Content model: this element does not have any contents.

2.19.1.  ‘target’ attribute (mandatory)

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

2.19.2.  ‘to’ attribute (mandatory)

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

2.20.  <dl>

A definition list. Each entry has a pair of elements: a term (<dt>)
and a definition (<dd>). (This is slightly different than the model
used in HTML, which allows for multiple terms for a single
definition.)

This element appears as a child element of: <abstract> (Section 2.1),
<aside> (Section 2.6), <blockquote> (Section 2.10), <dd>
(Section 2.18), <li> (Section 2.29), <note> (Section 2.33), <section>
(Section 2.46), <td> (Section 2.56), and <th> (Section 2.58).

Content model:

One or more sequences of:
1. One <dt> element

2. One <dd> element

2.20.1. ‘anchor’ attribute

Document-wide unique identifier for the list.

2.20.2. ‘hanging’ attribute

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

Allowed values:

- "false"
- "true" (default)

2.20.3. ‘spacing’ attribute

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

Allowed values:

- "normal" (default)
- "compact"

2.21. <dt>

The term being defined in a definition list.

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

Content model:

In any order:

- Text

- <bcp14> elements (Section 2.9)


2.21.1. ‘anchor’ attribute

Document-wide unique identifier for this term.

2.22. <em>

Indicates text that is semantically emphasized. This element will be displayed as italic after processing. This element can be combined with other character formatting elements, and the formatting will be additive.

This element appears as a child element of: <annotation> (Section 2.3), <blockquote> (Section 2.10), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <li> (Section 2.29), <preamble> (Section 3.6), <refcontent> (Section 2.39), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <tt> (Section 2.53), <td> (Section 2.56), <th> (Section 2.58), and <tt> (Section 2.62).

Content model:

In any order:

- Text

- <bcp14> elements (Section 2.9)
2.23.  <email>

Provides an email address.

The value is expected to be the addr-spec defined in Section 2 of
[RFC6068].

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

Content model: only text content.

2.23.1.  ‘ascii’ attribute

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

2.24.  <eref>

Represents an "external" link (as specified in the "target"
attribute).  This is useful for embedding URIs in the body of a
document.

If the <eref> element has non-empty text content, formatters should
use the content as the displayed text that is linked.  Otherwise the
formatter should use the value of the "target" attribute as the
displayed text.  Formatters will link the displayed text to the value
of the "target" attribute in a manner appropriate for the output
format.
For example, with an input of:

This is described at
<eref target="http://www.example.com/reports/r12.html"/>

An HTML formatter might generate

This is described at
<a href="http://www.example.com/reports/r12.html">

With an input of:

This is described
<eref target="http://www.example.com/reports/r12.html">
in this interesting report</eref>.

An HTML formatter might generate

This is described
<a href="http://www.example.com/reports/r12.html">
in this interesting report</a>.

This element appears as a child element of: <annotation>
(Section 2.3), <blockquote> (Section 2.10), <c> (Section 3.1), <cref>
(Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em>
(Section 2.22), <li> (Section 2.29), <name> (Section 2.32),
<postamble> (Section 3.5), <preamble> (Section 3.6), <strong>
(Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <tt>
(Section 2.53), <td> (Section 2.56), <th> (Section 2.58), <tt>
(Section 2.62), and <ttcol> (Section 3.9).

Content model: only text content.

2.24.1. ‘target’ attribute (mandatory)

URI of the link target (see Section 3 of [RFC3986]). This must begin
with a scheme name (such as "https://") and thus not be relative to
the URL of the current document.

2.25. <figure>

Contains a figure with a caption with the figure number. If the
element contains a <name> element, the caption will also show that
name.

This element appears as a child element of: <aside> (Section 2.6),
<blockquote> (Section 2.10), <dd> (Section 2.18), <li>
(Section 2.29), <section> (Section 2.46), <td> (Section 2.56), and <th> (Section 2.58).

Content model:

In this order:

1. One optional <name> element (Section 2.32)
2. Optional <iref> elements (Section 2.27)
3. One optional <preamble> element (Section 3.6)
4. In any order, but at least one of:

   * <artwork> elements (Section 2.5)
   * <sourcecode> elements (Section 2.48)

5. One optional <postamble> element (Section 3.5)

2.25.1. ‘align’ attribute

   Deprecated.

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

   Allowed values:

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

2.25.2. ‘alt’ attribute

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

2.25.3. ‘anchor’ attribute

   Document-wide unique identifier for this figure.
2.25.4. ‘height’ attribute

Deprecated.

2.25.5. ‘src’ attribute

Deprecated.

2.25.6. ‘suppress-title’ attribute

Deprecated. Figures always now get captions.

Allowed values:

- "true"
- "false" (default)

2.25.7. ‘title’ attribute

Deprecated. Use <name> instead.

2.25.8. ‘width’ attribute

Deprecated.

2.26. <front>

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

A <front> element may have more than one <seriesInfo> elements. A <seriesInfo> element determines the document number (for RFCs) or name (for Internet-Drafts). Another <seriesInfo> element determines the "maturity level" (see Section 4 of [RFC2026]), using values of "std" for "Standards Track", "bcp" for "BCP", "info" for "Informational", "exp" for "Experimental", and "historic" for "Historic". The "name" attributes of those multiple <seriesInfo> elements interact as described in the section on <seriesInfo>.

This element appears as a child element of: <reference> (Section 2.40), and <rfc> (Section 2.45).

Content model:

In this order:
1. One <title> element (Section 2.60)
2. One or more <author> elements (Section 2.7)
3. One optional <date> element (Section 2.17)
4. Optional <area> elements (Section 2.4)
5. Optional <workgroup> elements (Section 2.65)
6. Optional <keyword> elements (Section 2.28)
7. One optional <abstract> element (Section 2.1)
8. Optional <seriesInfo> elements (Section 2.47)
9. Optional <note> elements (Section 2.33)
10. One optional <boilerplate> element (Section 2.11)

2.27. <iref>

Provides terms for the document’s index.

Index entries can be either be regular entries (when just the "item" attribute is given) or nested entries (by specifying "subitem" as well), grouped under a regular entry.

Index entries generally refer to the exact place where the <iref> element occurred. An exception is the occurrence as a child element of <section>, in which case the whole section is considered to be relevant for that index entry. In some formats, index entries of this type might be displayed as range.

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

This element appears as a child element of: <annotation> (Section 2.3), <aside> (Section 2.6), <blockquote> (Section 2.10), <c> (Section 3.1), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <figure> (Section 2.25), <li> (Section 2.29), <postamble> (Section 3.5), <preamble> (Section 3.6), <section> (Section 2.46), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <table> (Section 2.54), <td> (Section 2.56), <th> (Section 2.58), <tt> (Section 2.62), and <ttcol> (Section 3.9).

Content model: this element does not have any contents.
2.27.1. ‘item’ attribute (mandatory)

The item to include.

2.27.2. ‘primary’ attribute

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

Allowed values:
- "true"
- "false" (default)

2.27.3. ‘subitem’ attribute

The subitem to include.

2.28. <keyword>

Specifies a keyword applicable to the document.

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

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

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

Content model: only text content.

2.29. <li>

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

This element appears as a child element of: <ol> (Section 2.34), and <ul> (Section 2.63).

Content model:

Either:

In any order, but at least one of:
* <artwork> elements (Section 2.5)
* <dl> elements (Section 2.20)
* <figure> elements (Section 2.25)
* <ol> elements (Section 2.34)
* <sourcecode> elements (Section 2.48)
* <t> elements (Section 2.53)
* <ul> elements (Section 2.63)

Or:

In any order, but at least one of:

* Text
* <bcp14> elements (Section 2.9)
* <cref> elements (Section 2.16)
* <em> elements (Section 2.22)
* <eref> elements (Section 2.24)
* <iref> elements (Section 2.27)
* <relref> elements (Section 2.44)
* <strong> elements (Section 2.50)
* <sub> elements (Section 2.51)
* <sup> elements (Section 2.52)
* <tt> elements (Section 2.62)
* <xref> elements (Section 2.66)
2.29.1. ‘anchor’ attribute

Document-wide unique identifier for this list item.

2.30. <link>

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

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

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

- The DOI for this document. The value for the "rel" attribute is "describedBy". The link should use the form specified in [DOI].

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

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

In RFC production mode, the prep tool needs to check the values for <link> before an RFC is published. In draft production mode, the prep tool might remove some <link> elements during the draft submission process.

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

Content model: this element does not have any contents.

2.30.1. ‘href’ attribute (mandatory)

The URI of the external document.

2.30.2. ‘rel’ attribute

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

Represents the main content of the document.

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

Content model:
One or more <section> elements (Section 2.46)

2.32. <name>

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

This element appears as a child element of: <figure> (Section 2.25), <note> (Section 2.33), <references> (Section 2.42), <section> (Section 2.46), <table> (Section 2.54), and <texttable> (Section 3.8).

Content model:
In any order:
  o  Text
  o  <cref> elements (Section 2.16)
  o  <eref> elements (Section 2.24)
  o  <relref> elements (Section 2.44)
  o  <tt> elements (Section 2.62)
  o  <xref> elements (Section 2.66)

2.33. <note>

Creates an unnumbered section that appears after the abstract.

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

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

Content model:
In this order:

1. One optional <name> element (Section 2.32)

2. In any order, but at least one of:

   * <dl> elements (Section 2.20)
   * <ol> elements (Section 2.34)
   * <t> elements (Section 2.53)
   * <ul> elements (Section 2.63)

2.33.1. ‘removeInRFC’ attribute

   If set to "true", this note is marked in the prep tool with text indicating that it should be removed before the document is published as an RFC.

   Allowed values:
   
   o "true"
   
   o "false" (default)

2.33.2. ‘title’ attribute

   Deprecated. Use <name> instead.

2.34. <ol>

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

   This element appears as a child element of: <abstract> (Section 2.1), <aside> (Section 2.6), <blockquote> (Section 2.10), <dd> (Section 2.18), <li> (Section 2.29), <note> (Section 2.33), <section> (Section 2.46), <td> (Section 2.56), and <th> (Section 2.58).

   Content model:

   One or more <li> elements (Section 2.29)
2.34.1. ‘anchor’ attribute

Document-wide unique identifier for the list.

2.34.2. ‘group’ attribute

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

2.34.3. ‘spacing’ attribute

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

Allowed values:

- "normal" (default)
- "compact"

2.34.4. ‘start’ attribute

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

2.34.5. ‘type’ attribute

The type of the labels on list items. If the length of the type 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, ...)

For type "a" and "A", after the 26th entry, the numbering starts at "aa"/"AA", then "ab"/"AB", and so on.

If the length of the type 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. Only one percent encoding other than "%%" is allowed in a type string.

It is an error for the type string to be empty. For bulleted lists, you use the <ul> element. For lists that have neither bullets nor numbers, use the <ul> element with the 'empty="true"' attribute.

If no type attribute is given, the default type is the same as "type='\%d.'".

2.35. <organization>

Specifies the affiliation of an author.

This information appears in both the "Author's Address" section and on the front page (see [RFC7322] for more information). If the value is long, an abbreviated variant can be specified in the "abbrev" attribute.

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

Content model: only text content.

2.35.1. 'abbrev' attribute

Abbreviated variant.
2.35.2. ‘ascii’ attribute

The ASCII equivalent of the organization’s name.

2.36. <phone>

Represents a phone number.

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

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

Content model: only text content.

2.37. <postal>

Contains optional child elements providing postal information. These elements will be displayed in an order that is specific to formatters. 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 a child element of: <address> (Section 2.2).

Content model:

Either:

In any order:

* <city> elements (Section 2.13)
* <code> elements (Section 2.14)
* <country> elements (Section 2.15)
* <region> elements (Section 2.43)
* <street> elements (Section 2.49)

Or:
2.38. `<postalLine>`

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

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

Content model: only text content.

2.38.1. ‘ascii’ attribute

The ASCII equivalent of the text in the address line.

2.39. `<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:

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

would render as:

```
```

This element appears as a child element of: `<reference>` (Section 2.40).

Content model:

In any order:

- Text

- `<bcp14>` elements (Section 2.9)
2.40.  <reference>

Represents a bibliographical reference.

This element appears as a child element of: <referencegroup> (Section 2.41), and <references> (Section 2.42).

Content model:

In this order:

1.  One <front> element (Section 2.26)

2.  In any order:

   *  <annotation> elements (Section 2.3)
   *  <format> elements (Section 3.3)
   *  <refcontent> elements (Section 2.39)
   *  <seriesInfo> elements (Section 2.47; deprecated in this context)

2.40.1.  ‘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.

2.40.2.  ‘quoteTitle’ attribute

Specifies whether or not the title in the reference should be quoted. This can be used to prevent quoting, such as on errata.

Allowed values:
2.40.3. ‘target’ attribute

Holds the URI for the reference.

2.41. <referencegroup>

Represents a list of bibliographic references that will be represented as a single reference. This is most often used for references in the STD and BCP series, where a single reference (such as "BCP 9") encompasses more than one RFC.

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

Content model:

One or more <reference> elements (Section 2.40)

2.41.1. ‘anchor’ attribute (mandatory)

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

2.42. <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 [RFC7322]). This vocabulary supports the split with the <name> child element. In general, the title should be either "Normative References" or "Informative References".

By default, the order of references is significant. Some formatters, however, might be able to be requested to sort them based on their anchor names.

This element appears as a child element of: <back> (Section 2.8).

Content model:
In this order:

1. One optional <name> element (Section 2.32)

2. In any order:
   * <reference> elements (Section 2.40)
   * <referencegroup> elements (Section 2.41)

2.42.1. 'anchor' attribute

   An optional user-supplied anchor for this section.

2.42.2. 'title' attribute

   Deprecated. Use <name> instead.

2.43. <region>

   Provides the region name in a postal address.

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

   Content model: only text content.

2.43.1. 'ascii' attribute

   The ASCII equivalent of the region name.

2.44. <relref>

   A relative link to a reference from the References section.
   Formatters that have links (such as HTML and PDF) are likely to render <relref> elements as external hyperlinks to the specified part of the reference, creating the link target by combining the base URI from the <reference> element with the "relative" attribute from this element. The "target" attribute is required, and it must be the anchor of a <reference> element.

   Either the "relative" or the "section" attribute must be present, but both cannot be given for a <relref> element. If a reference is an RFC or Internet-Draft that is in the v3 format and the desired relative reference is to a section of that reference, the "section" attribute is easier to use than the "relative" attribute because the value of "section" is just a section string such as "2.3".
An example of the <relref> element with text content might be:

See <relref section="2.3" target="RFC7878">the protocol overview</relref>
for more information.

An HTML formatter might generate:

See
<a href="http://www.rfc-editor.org/info/rfc7878#s-2.3">the protocol overview</a>
for more information.

This element appears as a child element of: <annotation>
(Section 2.3), <blockquote> (Section 2.10), <cref> (Section 2.16),
<dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <li>
(Section 2.29), <name> (Section 2.32), <preamble> (Section 3.6),
<strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52),
<t> (Section 2.53), <td> (Section 2.56), <th> (Section 2.58), and
<tt> (Section 2.62).

Content model: only text content.

2.44.1. ‘displayFormat’ attribute

This attribute is used to signal formatters what the desired format
of the relative reference should be. Formatters for document types
that have linking capability should wrap each part of the displayed
text in hyperlinks. If there is content in the <relref> element,
formatters will ignore the value of this attribute.

"of"

A formatter should display the relative reference as the contents
of the "derivedRemoteContent" attribute followed by a space, the
word "of", another space, and the value from the "target"
attribute enclosed in square brackets.

For example, if Section 2.3 of RFC 7878 has the title "Protocol
Overview", with an input of:

See
<relref section="2.3" target="RFC7878" displayFormat="of"/>
for an overview.

An HTML formatter might generate:
See
<a href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a> of
[RFC7878] for an overview.

Note that "displayFormat='of'" is the default for <relref> so it
does not need to be given in a <relref> element if that format is
desired.

"comma"

A formatter should display the relative reference as the value
from the "target" attribute enclosed in square brackets, a comma,
a space, and the "derivedRemoteContent" attribute.

For example, if Section 2.3 of RFC 7878 has the title "Protocol
Overview", with an input of:

See
<relref section="2.3" target="RFC7878" displayFormat="comma"/>,
for an overview.

An HTML formatter might generate:

See
[RFC7878],
<a href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a>, for an overview.

"parens"

A formatter should display the relative reference as the value
from the "target" attribute enclosed in square brackets, a space,
a left parenthesis, the "derivedRemoteContent" attribute, and a
right parenthesis.

For example, if Section 2.3 of RFC 7878 has the title "Protocol
Overview", with an input of:

See
<relref section="2.3" target="RFC7878" displayFormat="parens"/>
for an overview.

An HTML formatter might generate
"bare"

A formatter should display the relative reference as the contents of the "derivedRemoteContent" attribute and nothing else. This is useful when there are multiple relative references to a single base reference.

For example:

See
<relref section="2.3" target="RFC7878" displayFormat="bare"/>
and
<relref section="2.4" target="RFC7878" displayFormat="of"/>
for an overview.

An HTML formatter might generate:

See
<a href="http://www.rfc-editor.org/info/rfc7878#s-2.3">Section 2.3</a>
and
<a href="http://www.rfc-editor.org/info/rfc7878#s-2.4">Section 2.4</a> of
<a href="#RFC7878">RFC7878</a> for an overview.

Allowed values:

- "of" (default)
- "comma"
- "parens"
- "bare"

2.44.2. ‘relative’ attribute

Specifies a relative reference from the URI in the target reference. This value must include whatever leading character is needed to create the relative reference; typically, this is "#" for HTML documents.
2.44.3. ‘section’ attribute

Specifies a section of the target reference. If the reference is not an RFC or Internet-Draft, it is an error.

2.44.4. ‘target’ attribute (mandatory)

The anchor of the reference for this element. If this value is not an anchor to a <reference> or <referencegroup> element, it is an error. If the reference at the target has no URI, it is an error.

2.45. <rfc>

This is the root element of the xml2rfc vocabulary.

Content model:

In this order:

1. Optional <link> elements (Section 2.30)
2. One <front> element (Section 2.26)
3. One <middle> element (Section 2.31)
4. One optional <back> element (Section 2.8)

2.45.1. ‘category’ attribute

Deprecated; instead, use the "name" attribute in <seriesInfo>.

2.45.2. ‘consensus’ attribute

Affects the generated boilerplate. Note that the values of "no" and "yes" are deprecated and are replaced by "false" (the default) and "true".

See [RFC5741] for more information.

Allowed values:

- "no"
- "yes"
- "false" (default)
2.45.3. ‘docName’ attribute

Deprecated; instead, use the "value" attribute in <seriesInfo>.

2.45.4. ‘indexInclude’ attribute

Specifies whether or not a formatter is requested to include an index in generated files. If the source file has no <iref> elements, an index is never generated. This option is useful for generating documents where the source document has <iref> elements but the author no longer wants an index.

Allowed values:

- "true" (default)
- "false"

2.45.5. ‘ipr’ attribute

Represents the Intellectual Property status of the document. See Appendix A.1 for details.

Allowed values:

- "full2026"
- "noDerivativeWorks2026"
- "none"
- "full3667"
- "noModification3667"
- "noDerivatives3667"
- "full3978"
- "noModification3978"
- "noDerivatives3978"
- "trust200811"
o "noModificationTrust200811"

o "noDerivativesTrust200811"

o "trust200902"

o "noModificationTrust200902"

o "noDerivativesTrust200902"

o "pre5378Trust200902"

2.45.6. ‘iprExtract’ attribute

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

2.45.7. ‘number’ attribute

Deprecated; instead, use the "value" attribute in <seriesInfo>.

2.45.8. ‘obsoletes’ attribute

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

The prep tool will parse the attribute value so that incorrect references can be detected.

2.45.9. ‘prepTime’ attribute

The date that the XML was processed by a preparation tool. This is included in the XML file just before it is saved to disk. The value is formatted using the format from [RFC3339].

2.45.10. ‘seriesNo’ attribute

Deprecated; instead, use the "value" attribute in <seriesInfo>.

2.45.11. ‘sortRefs’ attribute

Specifies whether or not the prep tool will sort the references in each reference section.

Allowed values:
2.45.12. ‘submissionType’ attribute

The document stream.

See Section 2 of [RFC5741] for details.

Allowed values:

- "IETF" (default)
- "IAB"
- "IRTF"
- "independent"

2.45.13. ‘symRefs’ attribute

Specifies whether or not a formatter is requested to use symbolic references (such as "[RFC2119]"). If the value for this is "false", the references come out as numbers (such as "[3]").

Allowed values:

- "true" (default)
- "false"

2.45.14. ‘tocDepth’ attribute

Specifies number of levels of heading that formatter is requested to include in the table of contents; the default is "3".

2.45.15. ‘tocInclude’ attribute

Specifies whether or not a formatter is requested to include a table of contents in generated files.

Allowed values:

- "true" (default)
- "false"
2.45.16. ‘updates’ attribute

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

The prep tool will parse the attribute value so that incorrect references can be detected.

2.45.17. ‘version’ attribute

Specifies the version of xml2rfc syntax used in this document. The only expected value (for now) is "3".

2.46. <section>

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

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

This element appears as a child element of: <back> (Section 2.8), <boilerplate> (Section 2.11), <middle> (Section 2.31), and <section> (Section 2.46).

Content model:

In this order:

1. One optional <name> element (Section 2.32)

2. In any order:

* <artwork> elements (Section 2.5)
* <aside> elements (Section 2.6)
* <blockquote> elements (Section 2.10)
* <dl> elements (Section 2.20)
* <figure> elements (Section 2.25)
* <iref> elements (Section 2.27)
* <ol> elements (Section 2.34)
* <sourcecode> elements (Section 2.48)
* <t> elements (Section 2.53)
* <table> elements (Section 2.54)
* <texttable> elements (Section 3.8)
* <ul> elements (Section 2.63)

3. Optional <section> elements (Section 2.46)

2.46.1. ‘anchor’ attribute

Document-wide unique identifier for this section.

2.46.2. ‘numbered’ attribute

If set to "false", the formatter is requested to not display a section number. The prep tool 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:

- "true" (default)
- "false"

2.46.3. ‘removeInRFC’ attribute

If set to "true", the formatter is requested to mark this section with a paragraph at the beginning of the section indicating that it should be removed before the document is published as an RFC.

Allowed values:

- "true"
- "false" (default)

2.46.4. ‘title’ attribute

Deprecated. Use <name> instead.
2.46.5. ‘toc’ attribute

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

Allowed values:
- "include"
- "exclude"
- "default" (default)

2.47. <seriesInfo>

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

A processing tool determines if it is working on an RFC or an Internet-Draft by inspecting the "name" attribute of a <seriesInfo> element inside the <front> element inside the <rfc> element, looking for "rfc" or "Internet-Draft". (Specifying neither value in any of the <seriesInfo> elements can be useful for producing other types of documents, but is out-of-scope for this specification.)

It is invalid to have multiple <seriesInfo> elements inside the <front> element inside the <rfc> element that have "name" values of "rfc", or that have "name" values of "Internet-Draft", or that have both "rfc" and "Internet-Draft".

This element appears as a child element of: <front> (Section 2.26), and <reference> (Section 2.40; deprecated in this context).

Content model: this element does not have any contents.

2.47.1. ‘asciiName’ attribute

The ASCII equivalent of the name field.
2.47.2. ‘asciiValue’ attribute

The ASCII equivalent of the value field.

2.47.3. ‘name’ attribute (mandatory)

The name of the series.

Some of the values for “name” interact as follows:

- If a <front> element contains a <seriesInfo> element with a name of "Internet-Draft", it can also have at most one additional <seriesInfo> element with a name of "std", "bcp", "fyi", "exp", or "historic" to indicate the intended status of this Internet Draft, if it were to be later published as an RFC. If such an additional <seriesInfo> element has one of those names, the value attribute for that name MUST be "".

- If a <front> element contains a <seriesInfo> element with a name of "rfc", it can also have at most one additional <seriesInfo> element with a name of "std", "bcp", or "fyi" to indicate the current status of this RFC. If such an additional <seriesInfo> element has one of those names, the value attribute for that name MUST be the number within that series. That <front> element might also contain an additional <seriesInfo> with the name "exp" or "historic" and a value of "" to indicate the status of the RFC.

- A <front> element that has a <seriesInfo> element that has the name "Internet-Draft" MUST NOT also have a <seriesInfo> element that has the name "rfc".

- The DOI for the referenced document. This MUST NOT be used when <seriesInfo> element is an eventual child element of a <rfc> element, only as an eventual child of a <reference> element. The value attribute should use the form specified in [DOI].

Other values might be added at a later time by the RFC Editor.

2.47.4. ‘value’ attribute (mandatory)

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

For BCPs, FYIs, RFCs, and STDs this is the number within the series. For Internet-Drafts, it is the full draft name (ending with the two-digit version number). For DOIs, the value is given such as "10.123456/rfc1149", (the actual value will be specified later in [DOI]).
The name in the value should be the document name without any file
extension. For Internet Drafts, the value for this attribute should
be "draft-ietf-somewg-someprotocol-07", not
"draft-ietf-somewg-someprotocol-07.txt".

2.48. <sourcecode>

This element allows the inclusion of sourcecode into the document.

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

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

A common problem authors have with <sourcecode> is that the XML
processor returns errors if the text in the artwork contains either
the "&" or "<" character, or the string "]]>". To avoid these
problems, the "&" and "<" characters may be escaped using the strings
"&amp;" and "&lt;", respectively; the "]]>" string can be represented
as "]]>". Alternatively, they may be surrounded in a CDATA
structure: "<![CDATA[]]">". For example:

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

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

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

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

Output formatters that do pagination should attempt to keep source
code on a single page. This is to prevent source code that is split
across pages from looking like two separate pieces of code.

This element appears as a child element of: <blockquote>
(Section 2.10), <dd> (Section 2.18), <figure> (Section 2.25), <li>
2.48.1. ‘anchor’ attribute

Document-wide unique identifier for this sourcecode.

2.48.2. ‘name’ attribute

A filename suitable for the contents (such as for extraction to a local file). This attribute can be helpful for other kinds of tools (such as automated syntax checkers which work by extracting the source code). Note that the "name" attribute does not need to be unique for artwork elements in a document. If multiple sourcecode elements have the same name attribute, a formatter might assume that the elements are all fragments of a single file, and such a formatter can collect those fragments for later processing.

2.48.3. ‘src’ attribute

The URI reference of a source file ([RFC3986]).

It is an error to have both a "src" attribute and content in the <sourcecode> element.

2.48.4. ‘type’ attribute

Specifies the type of the sourcecode. The value of this attribute is free text with certain values designated as preferred.

The preferred values for <sourcecode> types are:

  o abnf
  o asn.1
  o bash
  o c++
  o c
  o cbor
  o dtd
The RFC Editor will maintain a complete list of the preferred values on its web site, and that list is expected to be updated over time. Thus, a consumer of v3 XML should not cause a failure when it encounters an unexpected type.

2.49.  <street>

Provides a street address.

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

Content model: only text content.

2.49.1. ‘ascii’ attribute

The ASCII equivalent of the street address.

2.50.  <strong>

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

This element appears as a child element of: <annotation> (Section 2.3), <blockquote> (Section 2.10), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <li> (Section 2.29), <preamble> (Section 3.6), <refcontent> (Section 2.39), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <td> (Section 2.56), <th> (Section 2.58), and <tt>
2.51.  <sub>

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

This element appears as a child element of: <annotation> (Section 2.3), <blockquote> (Section 2.10), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <li> (Section 2.29), <preamble> (Section 3.6), <refcontent> (Section 2.39), <strong> (Section 2.50), <t> (Section 2.53), <td> (Section 2.56), <th> (Section 2.58), and <tt> (Section 2.62).

Content model:

In any order:
2.52. <sup>
Causes the text to be displayed as superscript, approximately half a
letter-height higher than normal text. This element can be combined
with other character formatting elements, and the formatting will be
additive.

This element appears as a child element of: <annotation>
(Section 2.3), <blockquote> (Section 2.10), <cref> (Section 2.16),
<dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <li>
(Section 2.29), <preamble> (Section 3.6), <refcontent>
(Section 2.39), <strong> (Section 2.50), <t> (Section 2.53), <td>
(Section 2.56), <th> (Section 2.58), and <tt> (Section 2.62).

Content model:
In any order:

  o Text
  o <bcp14> elements (Section 2.9)
  o <cref> elements (Section 2.16)
  o <em> elements (Section 2.22)
  o <eref> elements (Section 2.24)
2.53. <t>

Contains a paragraph of text.

This element appears as a child element of: <abstract> (Section 2.1), <aside> (Section 2.6), <blockquote> (Section 2.10), <dd> (Section 2.18), <li> (Section 2.29), <list> (Section 3.4), <note> (Section 2.33), <section> (Section 2.46), <td> (Section 2.56), and <th> (Section 2.58).

Content model:
In any order:
  o Text
  o <bcp14> elements (Section 2.9)
  o <cref> elements (Section 2.16)
  o <em> elements (Section 2.22)
  o <eref> elements (Section 2.24)
  o <iref> elements (Section 2.27)
  o <list> elements (Section 3.4)
  o <relref> elements (Section 2.44)
  o <spanx> elements (Section 3.7)
  o <strong> elements (Section 2.50)
  o <sub> elements (Section 2.51)
  o <sup> elements (Section 2.52)
2.53.1. ‘anchor’ attribute

Document-wide unique identifier for this paragraph.

2.53.2. ‘hangText’ attribute

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

2.53.3. ‘keepWithNext’ attribute

Acts as a hint to the output formatters that do pagination to do a best effort attempt to keep the paragraph with the next element, whatever that happens to be. For example, the HTML output @media print CSS might translate this to page-break-after: avoid. For PDF, the paginator could attempt to keep the paragraph with the next element. Note: this attribute is strictly a hint and not always actionable.

Allowed values:

- "false" (default)
- "true"

2.53.4. ‘keepWithPrevious’ attribute

Acts as a hint to the output formatters that do pagination to do a best effort attempt to keep the paragraph with the previous element, whatever that happens to be. For example, the HTML output @media print CSS might translate this to page-break-before: avoid. For PDF, the paginator could attempt to keep the paragraph with the previous element. Note: this attribute is strictly a hint and not always actionable.

Allowed values:

- "false" (default)
- "true"
2.54.  <table>

Contains a table with a caption with the table number. If the element contains a <name> element, the caption will also show that name.

Inside the <table> element is optionally a <thead> element to contain the rows that will be the table’s heading and optionally a <tfoot> element to contain the rows of the table’s footer. If the XML is converted to a representation that has page breaks (such as PDFs, or printed HTML), the header and footer are meant to appear on each page.

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

Content model:

In this order:

1. One optional <name> element (Section 2.32)
2. Optional <iref> elements (Section 2.27)
3. One optional <thead> element (Section 2.59)
4. One or more <tbody> elements (Section 2.55)
5. One optional <tfoot> element (Section 2.57)

2.54.1. ‘anchor’ attribute

Document-wide unique identifier for this table.

2.55.  <tbody>

A container for a set of body rows for a table.

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

Content model:

One or more <tr> elements (Section 2.61)

2.55.1. ‘anchor’ attribute

Document-wide unique identifier for the tbody.
2.56. <td>

A cell in a table row.

This element appears as a child element of: <tr> (Section 2.61).

Content model:

Either:

One or more <t> elements (Section 2.53)

Or:

In any order, but at least one of:

* Text
* <artwork> elements (Section 2.5)
* <bcp14> elements (Section 2.9)
* <br> elements (Section 2.12)
* <cref> elements (Section 2.16)
* <dl> elements (Section 2.20)
* <em> elements (Section 2.22)
* <eref> elements (Section 2.24)
* <figure> elements (Section 2.25)
* <iref> elements (Section 2.27)
* <ol> elements (Section 2.34)
* <relref> elements (Section 2.44)
* <sourcecode> elements (Section 2.48)
* <strong> elements (Section 2.50)
* <sub> elements (Section 2.51)
2.56.1. ‘align’ attribute

Controls whether the content of the cell appears left justified (default), centered, or right justified. Note that "center" or "right" probably only work well in cells with plain text; any other elements might make the contents render badly.

Allowed values:

o "left" (default)

o "center"

o "right"

2.56.2. ‘anchor’ attribute

Document-wide unique identifier for the cell.

2.56.3. ‘border’ attribute

The width of the border for this cell. The default is 0, meaning no border.

2.56.4. ‘colspan’ attribute

The number of columns that the cell to span. For example, setting "colspan='3'" indicates that the cell occupies the same horizontal space as three cells in the row above or below this one.

2.56.5. ‘rowspan’ attribute

The number of rows that the cell to span. For example, setting "rowspan='3'" indicates that the cell occupies the same vertical space as three rows.
2.57.  `<tfoot>`

A container for a set of footer rows for a table.

This element appears as a child element of: `<table>` (Section 2.54).

Content model:
One or more `<tr>` elements (Section 2.61)

2.57.1.  `anchor` attribute

Document-wide unique identifier for the `tfoot`.

2.58.  `<th>`

A cell in a table row. When rendered, this will normally come out in
boldface; other than that, there is no difference between this and
the `<td>` element.

This element appears as a child element of: `<tr>` (Section 2.61).

Content model:
Either:

One or more `<t>` elements (Section 2.53)

Or:

In any order, but at least one of:

* Text
* `<artwork>` elements (Section 2.5)
* `<bcp14>` elements (Section 2.9)
* `<br>` elements (Section 2.12)
* `<cref>` elements (Section 2.16)
* `<dl>` elements (Section 2.20)
* `<em>` elements (Section 2.22)
2.58.1. ‘align’ attribute

Controls whether the content of the cell appears left justified (default), centered, or right justified. Note that "center" or "right" probably only work well in cells with plain text; any other elements might make the contents render badly.

Allowed values:

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

2.58.2. ‘anchor’ attribute

Document-wide unique identifier for the row.

2.58.3. ‘border’ attribute

The width of the border for this cell. The default is 0, meaning no border.
2.58.4. ‘colspan’ attribute

The number of columns that the cell to span. For example, setting "colspan='3'" indicates that the cell occupies the same horizontal space as three cells in the row above or below this one.

2.58.5. ‘rowspan’ attribute

The number of rows that the cell to span. For example, setting "rowspan='3'" indicates that the cell occupies the same vertical space as three rows.

2.59. <thead>

A container for a set of header rows for a table.

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

Content model:

One or more <tr> elements (Section 2.61)

2.59.1. ‘anchor’ attribute

Document-wide unique identifier for the thead.

2.60. <title>

Represents the document title.

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

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

Content model: only text content.

2.60.1. ‘abbrev’ attribute

Specifies an abbreviated variant of the document title.

2.60.2. ‘ascii’ attribute

The ASCII equivalent of the title.
2.61. <tr>

A row of a table.

This element appears as a child element of: <tbody> (Section 2.55), <tfoot> (Section 2.57), and <thead> (Section 2.59).

Content model:

In any order, but at least one of:

- <td> elements (Section 2.56)
- <th> elements (Section 2.58)

2.61.1. ‘anchor’ attribute

Document-wide unique identifier for the row.

2.61.2. ‘border’ attribute

The width of the border for this row. The default is 0, meaning no border. If the cells in this row also have "border" attributes, those values override this value for those cells.

2.62. <tt>

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

This element appears as a child element of: <annotation> (Section 2.3), <blockquote> (Section 2.10), <cref> (Section 2.16), <dd> (Section 2.18), <dt> (Section 2.21), <em> (Section 2.22), <li> (Section 2.29), <name> (Section 2.32), <preamble> (Section 3.6), <refcontent> (Section 2.39), <strong> (Section 2.50), <sub> (Section 2.51), <sup> (Section 2.52), <t> (Section 2.53), <td> (Section 2.56), and <th> (Section 2.58).

Content model:

In any order:

- Text
- <bcp14> elements (Section 2.9)
2.63.  <ul>

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

This element appears as a child element of: <abstract> (Section 2.1),
<aside> (Section 2.6), <blockquote> (Section 2.10), <dd>
(Section 2.18), <li> (Section 2.29), <note> (Section 2.33), <section>
(Section 2.46), <td> (Section 2.56), and <th> (Section 2.58).

Content model:
One or more <li> elements (Section 2.29)

2.63.1.  'anchor' attribute

Document-wide unique identifier for the list.

2.63.2.  '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"
2.63.3. ‘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"

2.64. <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 a child element of: <address> (Section 2.2).

Content model: only text content.

2.65. <workgroup>

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

In Internet-Drafts, this is used in the upper left corner of the  
boilerplate, replacing the "Network Working Group" string.  
Formatting software can append the words "Working Group" or "Research  
Group", depending on the "submissionType" property on the <rfc>  
element (Section 2.45.12).

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

Content model: only text content.

2.66. <xref>

A reference to an anchor in this document.  Formatters that have  
links (such as HTML and PDF) are likely to render <xref> elements as  
internal hyperlinks.  This element is useful for referring to  
references in the References section, to specific sections of this  
document, to specific figures, and so on.  The "target" attribute is  
required.
This element appears as a child element of: `<annotation>` (Section 2.3), `<blockquote>` (Section 2.10), `<c>` (Section 3.1), `<cref>` (Section 2.16), `<dd>` (Section 2.18), `<dt>` (Section 2.21), `<em>` (Section 2.22), `<li>` (Section 2.29), `<name>` (Section 2.32), `<postamble>` (Section 3.5), `<preamble>` (Section 3.6), `<strong>` (Section 2.50), `<sub>` (Section 2.51), `<sup>` (Section 2.52), `<tt>` (Section 2.53), `<td>` (Section 2.56), `<th>` (Section 2.58), `<tt>` (Section 2.62), and `<ttcol>` (Section 3.9).

Content model: only text content.

2.66.1. 'format' attribute

This attribute signals to formatters what the desired format of the reference should be. Formatters for document types that have linking capability should wrap the displayed text in hyperlinks.

"counter"

The "derivedContent" attribute will contain just a counter. This is used for targets that are `<section>`, `<figure>`, `<table>`, or item in an ordered list. Using "format='counter'" where the target is any other type of element is an error.

For example, with an input of:

```xml
<section anchor="overview">Protocol Overview</section>

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

An HTML formatter might generate "See Section <a href="#overview">1.7</a> for an overview."

"default"

The "derivedContent" attribute will contain a text fragment that describes the referenced part completely, such as "XML" for a target that is a `<reference>`, or "Section 2" or "Table 4" for a target to a non-reference.

For example, with an input of:

```xml
<section anchor="overview">Protocol Overview</section>

See <xref target="overview"/> for an overview.
```
An HTML formatter might generate "See <a href="#overview">Section 1.7</a> for an overview.".

"none"

Deprecated.

"title"

If the target is a <reference> element, the "derivedContent" attribute will contain the name of the reference, extracted from the <title> child of the <front> child of the reference. Or, if the target element has a <name> child element, the "derivedContent" attribute will contain the text content of that <name> element concatenated with the text content of each descendant node of <name> (that is, stripping out all of the XML markup, leaving only the text). Or, if the target element does not contain a <name> child element, the "derivedContent" attribute will contain the name of the "anchor" attribute of that element with no other adornment.

Allowed values:

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

2.66.2. ‘pageno’ attribute

Deprecated.

Allowed values:

- "true"
- "false" (default)

2.66.3. ‘target’ attribute (mandatory)

Identifies the document component being referenced. The value needs to match the value of the "anchor" attribute of an element in the document, otherwise it is an error.
2.67. `<svg>` (in namespace http://www.w3.org/2000/svg)

This element holds SVG art. The use of SVG in Internet Drafts and RFCs is covered in much more detail in [SVGforRFCs].

This element appears as a child element of: `<artwork>` (Section 2.5).

Content model: this element does not have any contents.

2.67.1. ‘TheRealDefinition’ attribute (mandatory)

(This is just a placeholder until [SVGforRFCs] has complete RNG that can be incorporated by reference in this draft.)

3. Elements from v2 That Have Been Deprecated

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

3.1. `<c>`

Deprecated. Instead, use `<tr>`, `<td>`, and `<th>`.

This element appears as a child element of: `<texttable>` (Section 3.8).

Content model:

In any order:

- Text
- `<bcp14>` elements (Section 2.9)
- `<ceref>` elements (Section 2.16)
- `<em>` elements (Section 2.22)
- `<eref>` elements (Section 2.24)
- `<iref>` elements (Section 2.27)
- `<spanx>` elements (Section 3.7)
- `<strong>` elements (Section 2.50)
3.2. <facsimile>

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

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

Content model: only text content.

3.3. <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 a child element of: <reference> (Section 2.40).

Content model: this element does not have any contents.

3.3.1. 'octets' attribute

Deprecated.

3.3.2. 'target' attribute

Deprecated.

3.3.3. 'type' attribute (mandatory)

Deprecated.

3.4. <list>

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

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

Content model:
One or more <t> elements (Section 2.53)

3.4.1. ‘counter’ attribute

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

3.4.2. ‘hangIndent’ attribute

Deprecated. Use <dl> instead.

3.4.3. ‘style’ attribute

Deprecated.

3.5. <postamble>

Deprecated. Instead, use a regular paragraph after the figure or table.

This element appears as a child element of: <figure> (Section 2.25), and <texttable> (Section 3.8).

Content model:

In any order:

- Text
- <bcp14> elements (Section 2.9)
- <cref> elements (Section 2.16)
- <em> elements (Section 2.22)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.27)
- <spanx> elements (Section 3.7)
- <strong> elements (Section 2.50)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)
3.6. <preamble>

Deprecated. Instead, use a regular paragraph before the figure or table.

This element appears as a child element of: <figure> (Section 2.25), and <texttable> (Section 3.8).

Content model:

In any order:

- Text
- <bcp14> elements (Section 2.9)
- <cref> elements (Section 2.16)
- <em> elements (Section 2.22)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.27)
- <spanx> elements (Section 3.7)
- <strong> elements (Section 2.50)
- <sub> elements (Section 2.51)
- <sup> elements (Section 2.52)
- <tt> elements (Section 2.62)
- <xref> elements (Section 2.66)

3.7. <spanx>

Deprecated.

This element appears as a child element of: <annotation> (Section 2.3), <c> (Section 3.1), <postamble> (Section 3.5), <preamble> (Section 3.6), and <t> (Section 2.53).
Content model: only text content.

3.7.1. ‘style’ attribute

Deprecated. Instead of `<span style="emph">`, use `<em>`; instead of `<span style="strong">`, use `<strong>`; instead of `<span style="verb">`, use `<tt>`.

3.7.2. ‘xml:space’ attribute

Deprecated.

Allowed values:

- "default"
- "preserve" (default)

3.8. `<texttable>`

Deprecated. Use `<table>` instead.

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

Content model:

In this order:

1. One optional `<name>` element (Section 2.32)
2. One optional `<preamble>` element (Section 3.6)
3. One or more `<ttcol>` elements (Section 3.9)
4. Optional `<c>` elements (Section 3.1)
5. One optional `<postamble>` element (Section 3.5)

3.8.1. ‘align’ attribute

Deprecated

Allowed values:

- "left"
3.8.2. 'anchor' attribute

Deprecated

3.8.3. 'style' attribute

Deprecated.

3.8.4. 'suppress-title' attribute

Deprecated.

Allowed values:

- "true"
- "false" (default)

3.8.5. 'title' attribute

Deprecated.

3.9. <ttcol>

Deprecated. Instead, use <tr>, <td>, and <th>.

This element appears as a child element of: <texttable> (Section 3.8).

Content model:

In any order:

- <cref> elements (Section 2.16)
- <eref> elements (Section 2.24)
- <iref> elements (Section 2.27)
- <xref> elements (Section 2.66)
- Text
3.9.1. 'align' attribute

Deprecated.

Allowed values:
- "left" (default)
- "center"
- "right"

3.9.2. 'width' attribute

Deprecated.

3.10. <vspace>

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

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

Content model: this element does not have any contents.

3.10.1. 'blankLines' attribute

Deprecated.

4. Internationalization Considerations

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

5. Security Considerations

The "name" attribute on the <artwork> element (Section 2.5.5) 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 "src" attribute on the <artwork> element can be used to read files from the local system. Processing tools must be careful to not
accept dangerous values for the filename, particularly those that contain absolute references outside the current directory.

The "type" attribute of the <artwork> and <sourcecode> elements is meant to encourage formatters to automatically extract known types of content from an RFC or Internet Draft. While extraction is probably safe, those tools 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 tool that is reading XML input needs to be suspicious of any content that it intends to post-process.

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

6.  IANA Considerations

6.1.  Internet Media Type Registration

IANA maintains the registry of Internet media types [BCP13] at <https://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 Section 9.1 of [RFC7303].
Encoding considerations:  Identical to those of "application/xml" as described in Section 9.1 of [RFC7303].
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 [RFC7303].
Interoperability considerations:  Different implementations of this format have had interoperability issues. It is not expected that publication of this application will cause those implementations to be fixed.
Published specification: This specification.

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

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

Additional information:

Deprecated alias names for this type: None.

Magic number(s): As specified for "application/xml" in Section 9.1 of [RFC7303].

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

Macintosh file type code(s): TEXT

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

Intended usage: COMMON

Restrictions on usage: None

Author: See Authors Section.

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

6.2. Link Relation Registration

The following is a proposed addition to [LINKRELATIONS].

Relation Name: convertedFrom

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

Reference: This document.

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

Application Data: none

7. Acknowledgments

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

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

8. References

8.1. Normative References


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


8.2. Informative References


January 2005,

[RFC5741] Daigle, L. and O. Kolkman, "RFC Streams, Headers,
and Boilerplates", RFC 5741, DOI 10.17487/RFC5741,
December 2009,

[RFC5743] Falk, A., "Definition of an Internet Research Task
Force (IRTF) Document Stream", RFC 5743,
DOI 10.17487/RFC5743, December 2009,

[RFC6068] Duerst, M., Masinter, L., and J. Zawinski, "The
'mailto' URI Scheme", RFC 6068, DOI 10.17487/
RFC6068, October 2010,

[RFC6266] Reschke, J., "Use of the Content-Disposition Header
Field in the Hypertext Transfer Protocol (HTTP)"
RFC 6266, DOI 10.17487/RFC6266, June 2011,

Requirements and Future Development", RFC 6949,
DOI 10.17487/RFC6949, May 2013,

[RFC7303] Thompson, H. and C. Lilley, "XML Media Types",
RFC 7303, DOI 10.17487/RFC7303, July 2014,

RFC 7322, DOI 10.17487/RFC7322, September 2014,

[RFCPOLICY] RFC Editor, "RFC Editorial Guidelines and
Procedures",

[RNC] Clark, J., "RELAX NG Compact Syntax", OASIS
committees/relax-ng/compact-20021121.html>.

Definition Languages (DSDL) - Part 2: Regular-
Grammar-Based Validation - RELAX NG. Second
A useful source of RNG-related information is <http://relaxng.org/>.


[UNICODE] The Unicode Consortium, "The Unicode Standard".

For the latest version, see <http://www.unicode.org/versions/latest/>.


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


Appendix A. Front Page (‘Boilerplate’) Generation

A.1. The /rfc/@ipr Attribute

This attribute value can take a long list of values, each of which describes an IPR policy for the document (Section 2.45.5). The values are not the result of a grand design, but remain simply for historic reasons. Of these values, only a few are currently in use; all others are supported by 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 (Section 2.45.12) 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.1.1. Current Values: ‘*trust200902’

The name for these values refers to the "IETF TRUST Legal Provisions Relating to IETF Documents", sometimes simply called the "TLP", which 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 prep tool automatically produces the "correct" text depending on the document’s date information (see above):

<table>
<thead>
<tr>
<th>TLP</th>
<th>starting with publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td>[TLP3.0]</td>
<td>2009-11-01</td>
</tr>
<tr>
<td>[TLP4.0]</td>
<td>2010-04-01</td>
</tr>
</tbody>
</table>
A.1.1.1. trust200902

This value should be used 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.1.1.2. 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.1.1.3. 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.1.1.4. pre5378Trust200902

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

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

See Section 4 of <http://trustee.ietf.org/docs/IETF-Copyright-FAQ.pdf> for further information about when to use this value.
A.1.2. Historic Values

A.1.2.1. Historic Values: 

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

A.1.2.2. Historic Values: 

"3978"
The attribute values "full3978", "noModification3978" and "noDerivatives3978" are similar to their counterparts above, except that they use text specified in Section 5 of [RFC3978].

A.1.2.3. Historic Values: 

"3667"
The attribute values "full3667", "noModification3667" and "noDerivatives3667" are similar to their counterparts above, except that they use text specified in Section 5 of [RFC3667].

A.1.2.4. Historic Values: 

"2026"
The attribute values "full2026" and "noDerivativeWorks2026" are similar to their counterparts above, except that they use text specified in Section 10 of [RFC2026].

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

A.2. The /rfc/@submissionType Attribute

The RFC Editor publishes documents from different "document streams", of which the "IETF stream" is the most prominent one. Other streams are the "independent stream" (used for things such as administrative information or April 1st RFCs), the "IAB stream" (Internet Architecture Board) and the "IRTF stream" (Internet Research Task Force).

The values for the attribute are "IETF" (the default value), "independent", "IAB", and "IRTF".

Historically, this attribute did not affect the final appearance of
RFCs, except for subtle differences in Copyright notices. Nowadays (as of [RFC5741]), the stream name appears in the first line of the front page, and it also affects the text in the "Status Of This Memo" section.

For current documents, setting submissionType attribute will have the following effect:

- For RFCs, the stream name appears in the upper left corner of the first page (in Internet Drafts, this is either "Network Working Group", or the value of the <workgroup> element).
- For RFCs, if affects the whole "Status Of This Memo" section (see Section 3.2.2 of [RFC5741]).
- For all RFCs and Internet Drafts, it determines whether the "Copyright Notice" mentions the Copyright on Code Components (see TLP, Section "Text To Be Included in IETF Documents").

A.3. The /rfc/@consensus Attribute

For some of the publication streams (see Appendix A.2), the "Status Of This Memo" section depends on whether there was a consensus to publish (again, see Section 3.2.2 of [RFC5741]).

The consensus attribute can be used to supply this information. The acceptable values are "true" (the default) and "false"; "yes" and "no" from v2 are deprecated. For documents in the IRTF stream, this value must be "false".

The effect of this value for the various streams is:

- "independent" and "IAB": none.
- "IETF": mention that there was an IETF consensus.
- "IRTF": the text needs to describe the consensus/review as described in Section 2.1 of [RFC5743].

Appendix B. The v3 Format and Processing Tools

This section describes topics that are specific to v3 processing tools. Note that there is some discussion of tools in the main body of the document as well. For example, some elements have descriptions of how a processing tool might create output from the element.

The expected design of the tools that will be used with v3 documents.
includes:

- A "prep tool" that takes a v3 document, makes many checks, adds and changes many attribute values, and creates a file that is a "prepared document". The prepared document is a valid v3 document. The prep tool is described in [PREPTOOL].

- The prep tool is expected to have many modes:
  
  * RFC mode -- The mode used by the RFC Editor to process the input from one of the RFC streams, and to process XML produced during the RFC editing process. The restrictions on the canonical XML for RFCs, as well as how the non-canonical formats will look, are described at <https://www.rfc-editor.org/rse/wiki/doku.php?id=design:format-and-content-rfcs>.
  
  * Draft mode -- The mode used by the Internet Draft submission tool. The restrictions for the XML from this mode will be described later.
  
  * Diagnostic mode -- A mode that can be used by document authors to look for errors or warnings before they submit their documents for publication.
  
  * Consolidation mode -- Produces output where no external resources are required to render the file output. This includes expanding the XInclude entities and DTD entities in place, and changing all elements that have "src" attributes with external links into either "data:" URI or content for the element, as specified in [PREPTOOL].

- Formatting tools that will create HTML, PDF, plain text, and possibly other output formats. These formatters will be created by the IETF, but others can create such tools as well. The IETF tools are expected to take prepared documents as input.

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

The prep tool is expected to have clear error reporting, giving more context than just a line number. For example, the error messages
should differentiate between errors in XML and those from the v3 format.

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

B.1. Including External Text with XInclude

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

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

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

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

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

<references>
</references>

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

<x:include href="/home/chris/ietf/drafts/commoncontext.xml"/>

In general, XInclude should be used instead of ENTITY references and
B.2. Anchors and IDs

People writing and reading Internet Drafts and RFCs often want to make reference to specific locations in those documents. In the case of RFC authors, it is common to want to reference another part of their document, such as "see Section 3.2 of this document". Readers, on the other hand, want to reference part of documents that they didn't write, such as "see Section 3.2 of RFC 6949". The XML vocabulary in this document attempts to support both sets of people.

Authors can leave anchors in a document that can later be used for references with the "anchor" attribute. Anchors can be included in the following elements: <artwork>, <aside>, <blockquote>, <cref>, <figure>, <li>, <reference>, <referencegroup>, <section>, <sourcecode>, <t>, and <table>. The author can then refer to that anchor in the "target" attribute of the <xref> element.

Readers can refer to any element that has an "anchor" attribute by that attribute. Note, however, that most of the time, elements won't have anchors. In the common case, the reader wants to refer to an element that does not have an "anchor" attribute, but that element has "pn" attribute.

Processing tools add the "pn" attribute to many elements during processing. This attribute and its value are automatically generated by the tool if the attribute is not there; if the attribute is already there, the tool may replace the value.

B.2.1. Overlapping Values

In the HTML representation of this XML vocabulary, both anchors and "pn" attributes will be used in the "id" attributes of elements. Thus, there can be no overlap between the names entered in "anchor" attributes, in "slugifiedName" attributes, and those that are generated for the "pn" attributes. Also, there are some values for the "anchor" values that are reserved for sections, and those sections can only have those anchor values.

The following rules prevent this overlap:

- "pn" for regular sections always has the format "s-nnn", where "nnn" is the section or appendix number. For example, this would be "s-2.1.3" for Section 2.1.3 and "s-a" for Appendix A. For the <abstract> element, it is always "s-abstract". For the <note> element, it is always "s-note-nnn", where "nnn" is a sequential value. For the <boilerplate> element, it is always
"s-boilerplate-nnn", where "nnn" is a sequential value.

- "pn" for <figure> elements always has the format "f-nnn", where "nnn" is the figure number. For example, this would be "f-5" for Figure 5.

- "pn" for <table> elements always has the format "t-nnn", where "nnn" is the table number. For example, this would be "t-5" for Table 5.

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

- "slugifiedName" always has the format "n-ttt", where "ttt" is the text of the name after slugification. For example, this would be "n-protocol-overview" for the name "Protocol Overview". The actual conversions done in slugification will be specified at a later time.

- Anchors must never overlap with any of the above. The easiest way to assure that is to not pick an anchor name that starts with a single letter followed by a hyphen. If an anchor does overlap with one of the types of names above, the processing tool will reject the document.

B.3. Attributes Controlled by the Prep Tool

Many elements in the v3 vocabulary have new attributes whose role is to hold values generated by the prep tool. These attributes can exist in documents that are input to the prep tool; however, any of these attributes might be added, removed, or changed by the prep tool. Thus, it is explicitly unsafe for a document author to include these attributes and expect that their values will survive processing by the prep tool.

The attributes that are controlled by the prep tool are:

- The "pn" attribute in any element -- The number for this item within the section. The numbering is shared with other elements of a section. The "pn" attribute is added to many block-level elements inside sections.

- <artwork> originalSrc -- This attribute is filled with the original value of the "src" attribute if that attribute is removed by the prep tool.
<iref> "irefid" attribute -- This attribute is filled with an identifier used when creating indexes.

<name> "slugifiedName" attribute -- This attribute is filled with a "slugified" version of the text in the element. This attribute can be used in the output formats for elements that have both names and numbers.

<relref> "derivedLink" attribute -- This attribute is filled with the link that is derived from combining the URI from the reference and the relative part that is either a copy of the "relative" attribute or a section number derived from the "section" attribute.

<relref> "derivedRemoteContent" attribute -- If the <relref> element has text content, this attribute is filled with that content; the "displayFormat" attribute is set to "bare" if that attribute is not already set. If the <relref> element has no text content, this attribute is filled with the text for the remote link, such as "Section 2.3" or "Table 5". The prep tool might determine this text by reading the target reference and, if it is a RFC or Internet-Draft in the v3 format, finding the anchor given in the "relative" attribute or derived from the "section" attribute, and using the title of that element. If the reference is not an RFC or Internet-Draft in the v3 format, the text fragment is simply the value of the "relative" or "section" attribute. This will rarely produce a good result in formatted output so, for these documents, the <relref> element should contain text content.

<rfc> "expiresDate" attribute -- This attribute is filled with the date that an Internet Draft expires. The date is in the format yyyy-mm-dd.

<rfc> "mode" attribute -- This attribute is filled with a string that indicates what mode the prep tool was in when it processed the XML, such as whether it was processing a file to become an Internet-Draft or an RFC.

<rfc> "scripts" attribute -- This attribute is filled with a list of scripts needed to render this document. The list is comma-separated, with no spaces allowed. The order is unimportant. The names come from [UAX24]. For example, if the document has Chinese characters in it, the value might be "Common,Latin,Han".

<sourcecode> "originalSrc" attribute -- This attribute is filled with the original value of the "src" attribute if that attribute is removed by the prep tool.
<xref> "derivedContent" attribute -- This attribute is filled in if there is no content in the <xref> element. The value for this attribute is based on the value in the "displayFormat" attribute. Examples of how this value is filled can be found at Section 2.66.1.

In addition, note that the contents of the <boilerplate> element is controlled by the prep tool.

Appendix C. Relax NG Schema

The following is the RelaxNG schema for the v3 format.

```xml
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
namespace ns1 = "http://www.w3.org/2000/svg"

# xml2rfc Version 3 grammar
rfc =
  element rfc {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute number { text }?,
    [ a:defaultValue = "" ] attribute obsoletes { text }?,
    [ a:defaultValue = "" ] attribute updates { text }?,
    attribute category { text }?,
    attribute mode { text }?,
    [ a:defaultValue = "false" ] attribute consensus { "no" | "yes" | "false" | "true" }?,
    attribute seriesNo { text }?,
    attribute ipr {
      "full2026"
      "noDerivativeWorks2026"
      "none"
      "full3667"
      "noModification3667"
      "noDerivatives3667"
      "full3978"
      "noModification3978"
      "noDerivatives3978"
      "trust200811"
      "noModificationTrust200811"
      "noDerivativesTrust200811"
      "trust200902"
      "noModificationTrust200902"
      "noDerivativesTrust200902"
      "pre5378Trust200902"
    }
  }
```
title =
    element title {
        attribute xml:base { text }?,
        attribute xml:lang { text }?,
        attribute abbrev { text }?,
        attribute ascii { text }?,
        text
    }

author =
    element author {
        attribute xml:base { text }?,
        attribute xml:lang { text }?,
        attribute initials { text }?,
        attribute asciiInitials { text }?,
        attribute surname { text }?,
        attribute asciiSurname { text }?,
        attribute fullname { text }?,
        attribute role { "editor" }?,
        attribute asciiFullName { text }?,
        organization?,
        address?
    }

organization =
    element organization {
        attribute xml:base { text }?,
        attribute xml:lang { text }?,
        attribute abbrev { text }?,
        attribute ascii { text }?,
        text
    }

address =
    element address {
        attribute xml:base { text }?,
        attribute xml:lang { text }?,
        postal?,
        phone?,
        facsimile?,
        email?,
        uri?
    }

postal =
    element postal {
attribute xml:base { text }?,
attribute xml:lang { text }?,
((city | code | country | region | street)* | postalLine+)
)

city =
element city {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute ascii { text }?,
    text
}

region =
element region {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute ascii { text }?,
    text
}

code =
element code {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute ascii { text }?,
    text
}

country =
element country {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute ascii { text }?,
    text
}

postalLine =
element postalLine {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    text
}
phone =
  element phone {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    text
  }

facsimile =
  element facsimile {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    text
  }

e-mail =
  element e-mail {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute ascii { text }?,
    text
  }

uri =
  element uri {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    text
  }

date =
  element date {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute day { text }?,
    attribute month { text }?,
    attribute year { text }?,
    empty
  }

area =
  element area {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    }
text
}

workgroup =
element workgroup {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  text
}

keyword =
element keyword {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  text
}

abstract =
element abstract {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  (dl | ol | t | ul)+
}

note =
element note {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute pn { text }?,
  attribute title { text }?,
  attribute pn { text }?,
  [ a:defaultValue = "false" ]
  attribute removeInRFC { "true" | "false" }?,
  name?,
  (dl | ol | t | ul)+
}

boilerplate =
element boilerplate {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute pn { text }?,
  section+
}

middle =
element middle {

section =
  element section {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    attribute title { text }?,
    [ a:defaultValue = "true" ]
    attribute numbered { "true" | "false" }?,
    [ a:defaultValue = "default" ]
    attribute toc { "include" | "exclude" | "default" }?,
    [ a:defaultValue = "false" ]
    attribute removeInRFC { "true" | "false" }?,
    name?,
    { artwork
      aside
      blockquote
      dl
      figure
      iref
      ol
      sourcecode
      t
      table
      texttable
      ul)*,
    section* }
}

name =
  element name {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute slugifiedName { text }?,
    { text | cref | eref | relref | tt | xref }*
  }

t =
  element t {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    }
attribute hangText { text }?,
  [ a:defaultValue = "false" ]
attribute keepWithNext { "false" | "true" }?,
  [ a:defaultValue = "false" ]
attribute keepWithPrevious { "false" | "true" }?,
  (text
    bcp14
cref
etem
eref
  )
}

aside =
element aside {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  (artwork | dl | figure | iref | \\list | ol | t | table | ul)*
}

blockquote =
element blockquote {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  attribute cite { text }?,
  attribute quotedFrom { text }?,
  ((artwork | dl | figure | ol | sourcecode | t | ul)+
    (text
      bcp14
cref
etem
eref
    )
  )
}
\list =
  element list {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    [ a:defaultValue = "empty" ] attribute style { text }?,
    attribute hangIndent { text }?,
    attribute counter { text }?,
    t+
  }

ol =
  element ol {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    [ a:defaultValue = "1" ] attribute type { text }?,
    [ a:defaultValue = "1" ] attribute start { text }?,
    attribute group { text }?,
    [ a:defaultValue = "normal" ] attribute spacing { "normal" | "compact" }?,
    li+
  }

ul =
  element ul {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    [ a:defaultValue = "normal" ] attribute spacing { "normal" | "compact" }?,
    [ a:defaultValue = "false" ] attribute empty { "false" | "true" }?,
    li+
  }

li =
  element li {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute
attribute pn { text }?,
  ((artwork | dl | figure | ol | sourcecode | t | ul)+
   | (text
     bcp14
cref	em
eref
iref
relref
strong
sub
sup
tt
xref)+)
}

dl =
element dl {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  [ a:defaultValue = "normal" ]
  attribute spacing { "normal" | "compact" }?,
  [ a:defaultValue = "true" ]
  attribute hanging { "false" | "true" }?,
  (dt, dd)+
}

dt =
element dt {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  (text
    bcp14
cref	em
eref
iref
relref
strong
sub
sup
tt
xref)*
}
dd =
element dd {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  ((artwork | dl | figure | ol | sourcecode | t | ul)+
  | (text
    bcp14
    cref
    em
    eref
    iref
    relref
    strong
    sub
    sup
    tt
    xref)+)
}

dxref =
element xref {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute target { xsd:IDREF },
  [ a:defaultValue = "false" ]
  attribute pageno { "true" | "false" }?,
  [ a:defaultValue = "default" ]
  attribute format { "default" | "title" | "counter" | "none" }?,
  attribute derivedContent { text }?,
  text
}

deref =
element relref {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute target { xsd:IDREF },
  [ a:defaultValue = "of" ]
  attribute displayFormat { "of" | "comma" | "parens" | "bare" }?,
  (attribute section { text },
    attribute relative { text }?),
  attribute derivedRemoteContent { text }?,
  attribute derivedLink { text }?,
  text
}
eref =
  element eref {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute target { text },
    text
  }

iref =
  element iref {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute item { text },
    [ a:defaultValue = "" ] attribute subitem { text }?,
    [ a:defaultValue = "false" ]
    attribute primary { "true" | "false" }?,
    [ a:defaultValue = "" ] attribute irefid { text }?,
    empty
  }

cref =
  element cref {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute source { text }?,
    [ a:defaultValue = "true" ]
    attribute display { "true" | "false" }?,
    (text | em | eref | relref | strong | sub | sup | tt | xref)*
  }

tt =
  element tt {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text
      bcp14
cref	em
eref
relref
strong
sub
sup
xref)*
  }
strong =
    element strong {
      attribute xml:base { text }?,
      attribute xml:lang { text }?,
      (text
       | bcp14
       | cref
       | em
       | eref
       | iref
       | relref
       | sub
       | sup
       | tt
       | xref)*)
    }

em =
    element em {
      attribute xml:base { text }?,
      attribute xml:lang { text }?,
      (text
       | bcp14
       | cref
       | em
       | eref
       | iref
       | relref
       | strong
       | sub
       | sup
       | tt
       | xref)*)
    }

sub =
    element sub {
      attribute xml:base { text }?,
      attribute xml:lang { text }?,
      (text
       | bcp14
       | cref
       | em
       | eref
       | iref
       | relref
       | strong
       | tt
       | xref)*)
sup =
  element sup {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text
      bcp14
      cref
      em
      iref
      relref
      strong
      tt
      xref)*
  }

spanx =
  element spanx {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    [ a:defaultValue = "preserve" ]
    attribute xml:space { "default" | "preserve" }?,
    [ a:defaultValue = "emph" ] attribute style { text }?,
    text
  }

vspace =
  element vspace {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    [ a:defaultValue = "0" ] attribute blankLines { text }?,
    empty
  }

figure =
  element figure {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    [ 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 }?,
name?,
iref*,
preamble?,
(artwork | sourcecode)+,
postamble?
}

table =
  element table {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    name?,
    iref*,
    thead?,
    tbody+,
    tfoot?
  }

preamble =
  element preamble {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text
      bcp14
      cref
      em
      eref
      iref
      relref
      spanx
      strong
      sub
      sup
      tt
      xref)*
  }

artwork =
  element artwork {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
  }
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 }?,
attribute originalSrc { text }?,
(text* | svg)
)

svg =
  element ns1:svg {
    [ a:defaultValue = "can be found in draft-brownlee-svg-rfc" ]
    attribute TheRealDefinition { text }*
  }

sourcecode =
  element sourcecode {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    [ a:defaultValue = "" ] attribute name { text }?,
    [ a:defaultValue = "" ] attribute type { text }?,
    attribute src { text }?,
    attribute originalSrc { text }?,
    text
  }

thead =
  element thead {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    tr+
  }

tbody =
  element tbody {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    tr+
  }
tfoot =
element tfoot {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    tr+
}

tr =
element tr {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    [ a:defaultValue = "0" ] attribute border { text }?,
    (td | th)+
}

td =
element td {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    [ a:defaultValue = "0" ] attribute border { text }?,
    [ a:defaultValue = "0" ] attribute colspan { text }?,
    [ a:defaultValue = "0" ] attribute rowspan { text }?,
    [ a:defaultValue = "left" ]
    attribute align { "left" | "center" | "right" }?,
    (t+ | (text
        artwork
        bcp14
        br
        cref
        dl
        em
        eref
        figure
        iref
        ol
        relref
        sourcecode
        strong
        sub
        sup
    )}
th =
  element th {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    [ a:defaultValue = "0" ] attribute border { text }?,
    [ a:defaultValue = "0" ] attribute colspan { text }?,
    [ a:defaultValue = "0" ] attribute rowspan { text }?,
    [ a:defaultValue = "left" ] attribute align { "left" | "center" | "right" }?,
    (text |
     artwork | bcp14 | br | cref | dl | em | eref | figure | iref | ol | relref | sourcecode | strong | sub | sup | tt | ul | xref)+)
  }

postamble =
  element postamble {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text | cref | eref | iref | spanx | xref)*
  }

texttable =
  element texttable {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    attribute border { text }?,
    attribute colspan { text }?,
    attribute rowspan { text }?,
    [ a:defaultValue = "0" ] attribute align { "left" | "center" | "right" }?,
    (text |
     artwork | bcp14 | br | cref | dl | em | eref | figure | iref | ol | relref | sourcecode | strong | sub | sup | tt | ul | xref)+
  }
attribute xml:lang { text }?,
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" }?,
name?,
preamble?,
ttcol+,
c*,
postamble?
}

ttcol =
element ttcol {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute width { text }?,
    [ a:defaultValue = "left" ]
    attribute align { "left" | "center" | "right" }?,
    (cref | eref | iref | xref | text)*
}

c =
element c {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text | cref | eref | iref | spanx | xref)*
}

bcp14 =
element bcp14 {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
text
}

br =
element br {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    empty
}

back =
element back {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  displayreference*,
  references*,
  section*
}

displayreference =
  element displayreference {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute target { xsd:IDREF },
    attribute to { text }
  }

references =
  element references {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute title { text }?,
    name?,
    (reference | referencegroup)*
  }

reference =
  element reference {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID },
    attribute target { text }?,
    [ a:defaultValue = "true" ]
    attribute quoteTitle { "true" | "false" }?,
    front,
    (annotation | format | refcontent | seriesInfo)*
  }

referencegroup =
  element referencegroup {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID },
    reference+
  }

seriesInfo =
  element seriesInfo {

attribute xml:base { text }?,
attribute xml:lang { text }?,
attribute name { text },
attribute value { text },
attribute asciiName { text }?,
attribute asciiValue { text }?,
empty
)

format =
  element format {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute target { text }?,
    attribute type { text },
    attribute octets { text }?,
    empty
  }

annotation =
  element annotation {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text
     | bcp14
     | cref
     | em
     |eref
     | iref
     | relref
     | spanx
     | strong
     | sub
     | sup
     | tt
     | xref)*
  }

refcontent =
  element refcontent {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    (text | bcp14 | em | strong | sub | sup | tt)*
  }

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.

```xml
namespace a = "http://relaxng.org/ns/compatibility/annotations/1.0"
+ namespace ns1 = "http://www.w3.org/2000/svg"

+ # xml2rfc Version 3 grammar
rfc =
  element rfc {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     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 category { text }?,
+     attribute mode { text }?,
+     [ a:defaultValue = "false" ]
+     attribute consensus { "no" | "yes" | "false" | "true" }?,
+     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 {
```

Hoffman                   Expires March 7, 2016               [Page 118]```
"IETF" | "IAB" | "IRTF" | "independent"
?,
attribute docName { text }?,
- [ a:defaultValue = "en" ] attribute xml:lang { text }?,
+ [ a:defaultValue = "false" ]
+ attribute sortRefs { "true" | "false" }?,
+ [ a:defaultValue = "true" ]
+ attribute symRefs { "true" | "false" }?,
+ [ a:defaultValue = "true" ]
+ attribute tocInclude { "true" | "false" }?,
+ [ a:defaultValue = "false" ]
+ attribute tocDepth { text }?,
+ [ a:defaultValue = "true" ]
+ attribute indexInclude { "true" | "false" }?,
+ [ a:defaultValue = "true" ]
+ attribute version { text }?,
+ [ a:defaultValue = "Common, Latin" ] attribute scripts { text }
+
+ attribute expiresDate { text }?,
+ link*,
+ link =
+ element link {
  + attribute xml:base { text }?,
  + attribute xml:lang { text }?,
  + attribute href { text },
  + attribute rel { text }?
  + }
front =
+ element front {
  + title, author+, date, area*, workgroup*, keyword*, abstract?,
  - note*
  + attribute xml:base { text }?,
  + attribute xml:lang { text }?,
  + title,
  + author+,
  + date?,
  + area*,
  + workgroup*,
  + keyword*,
  + abstract?,
  + seriesInfo*,
  + note*,
  + boilerplate?
  }
+ title =
element title {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute abbrev { text }?,
+     attribute ascii { text }?,
text
}
author =
  element author {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute initials { text }?,
+     attribute asciiInitials { text }?,
+     attribute surname { text }?,
+     attribute asciiSurname { text }?,
+     attribute fullname { text }?,
+     attribute role { "editor" }?,
+     attribute asciiFullName { text }?,
organization?,
address?
}
organization =
  element organization {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute abbrev { text }?,
+     attribute ascii { text }?,
+     text
+   } address =
+   element address {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     postal?,
+     phone?,
+     facsimile?,
+     email?,
+     uri?
+   }
+   element postal {
+     attribute xml:base { text }?,
+     ((city | code | country | region | street)* | postalLine+)
+   }
+   element street {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     postal?
+   }
+   element postal {
+     attribute xml:base { text }?,
+     ((city | code | country | region | street)* | postalLine+)
+   }
+   element street {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     postal?
+   }
+   attribute xml:lang { text }?,
+   attribute ascii { text }?,
+   text
+ )
+ city =
+   element city {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute ascii { text }?,
+     text
+ )
+ region =
+   element region {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute ascii { text }?,
+     text
+ )
+ code =
+   element code {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute ascii { text }?,
+     text
+ )
+ country =
+   element country {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute ascii { text }?,
+     text
+ )
+ postalLine =
+   element postalLine {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute ascii { text }?,
+     text
+ )
+ phone =
+   element phone {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     text
+ )
+ facsimile =
+   element facsimile {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     text
+ )
attribute xml:lang { text }?,
  text
+
+ email =
  + element email {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + attribute ascii { text }?,
    + text
    + }
+ uri =
  + element uri {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + text
    + }

+ address = element address { postal?, phone?, facsimile?, email?,
  uri? }
  + postal = element postal { street+, (city | region | code |
  country)* }
  + street = element street { text }
  + city = element city { text }
  + region = element region { text }
  + code = element code { text }
  + country = element country { text }
  + phone = element phone { text }
  + facsimile = element facsimile { text }
  + email = element email { text }
  + uri = element uri { text }
  date =
    + element date {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      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+ }
+ area =
  + element area {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + text
    + }

Hoffman                   Expires March 7, 2016               [Page 122]
+ workgroup =
+   element workgroup {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     text
+   }
+ + keyword =
+   element keyword {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     text
+   }
+ + abstract =
+   element abstract {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute anchor { xsd:ID }?,
+     attribute pn { text }?,
+     (dl | ol | t | ul)+
+   }
+ + note =
+   element note {
+     attribute title { text },
+     t+
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute title { text }?,
+     attribute pn { text }?,
+     [ a:defaultValue = "false" ]
+     attribute removeInRFC { "true" | "false" }?,
+     name?,
+     (dl | ol | t | ul)+
+   }
+ + boilerplate =
+   element boilerplate {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute pn { text }?,
+     section+
+   }
+ + middle =
+   element middle {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     section+
+   }
+ middle = element middle { section+ }
element section {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
-    attribute title { text },
+    attribute pn { text }?,
+    attribute title { text }?,
+    [ a:defaultValue = "true" ]
+    attribute numbered { "true" | "false" }?,
+    [ a:defaultValue = "default" ]
    attribute toc { "include" | "exclude" | "default" }?,
-    (t | figure | texttable | iref)*,
+    [ a:defaultValue = "false" ]
+    attribute removeInRFC { "true" | "false" }?,
+    name?,
+    (artwork
+     | aside
+     | blockquote
+     | dl
+     | figure
+     | iref
+     | ol
+     | sourcecode
+     | t
+     | table
+     | texttable
+     | ul)*,
  section*
+    name =
+    element name {
+        attribute xml:base { text }?,
+        attribute xml:lang { text }?,
+        attribute slugifiedName { text }?,
+        (text | cref | eref | relref | tt | xref)*
+    }
+}
+    t =
+    element t {
+        attribute xml:base { text }?,
+        attribute xml:lang { text }?,
        attribute anchor { xsd:ID }?,
+        attribute pn { text }?,
        attribute hangText { text }?,
+        [ a:defaultValue = "false" ]
+        attribute keepWithNext { "false" | "true" }?,
+        [ a:defaultValue = "false" ]
+        attribute keepWithPrevious { "false" | "true" }?,
        (text

aside =
  element aside {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    (artwork | dl | figure | iref | \list | ol | t | table | ul)*
  }
blockquote =
  element blockquote {
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute pn { text }?,
    attribute cite { text }?,
    attribute quotedFrom { text }?,
    ((artwork | dl | figure | ol | sourcecode | t | ul)+
     | (text
      bcp14
      cref
      em
     eref
    iref
    relref
    strong
    sub
    sup
    tt)
\list =
  element list {
    attribute style { text }?,
    attribute xml:base { text }?,
    attribute xml:lang { text }?,
    + [ a:defaultValue = "empty" ] attribute style { text }?,
    attribute hangIndent { text }?,
    attribute counter { text }?,
    t+
  }
+ ol =
+   element ol {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute anchor { xsd:ID }?,
+     attribute pn { text }?,
+     + [ a:defaultValue = "1" ] attribute type { text }?,
+     + [ a:defaultValue = "1" ] attribute start { text }?,
+     attribute group { text }?,
+     + [ a:defaultValue = "normal" ]
+     + attribute spacing { "normal" | "compact" }?,
+     li+
+   }
+ ul =
+   element ul {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute anchor { xsd:ID }?,
+     attribute pn { text }?,
+     + [ a:defaultValue = "normal" ]
+     + attribute spacing { "normal" | "compact" }?,
+     + [ a:defaultValue = "false" ]
+     + attribute empty { "false" | "true" }?,
+     li+
+   }
+ li =
+   element li {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute anchor { xsd:ID }?,
+     attribute pn { text }?,
+     + ((artwork | dl | figure | ol | sourcecode | t | ul)+
+     + (text
+       + bcp14
+       + cref
+       + em

Hoffman                   Expires March 7, 2016               [Page 126]
+      eref
+      iref
+      relref
+      strong
+      sub
+      sup
+      tt
+      xref)+(}
+ }  dl =
+ element dl {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
+    attribute anchor { xsd:ID }?,
+    [ a:defaultValue = "normal" ]
+    attribute spacing { "normal" | "compact" }?,
+    [ a:defaultValue = "true" ]
+    attribute hanging { "false" | "true" }?,
+    (dt, dd)+
+   }
+ dt =
+ element dt {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
+    attribute anchor { xsd:ID }?,
+    attribute pn { text }?,
+    (text
+     | bcp14
+     | cref
+     | em
+     | eref
+     | iref
+     | relref
+     | strong
+     | sub
+     | sup
+     | tt
+     | xref)*
+   }
+ dd =
+ element dd {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
+    attribute anchor { xsd:ID }?,
+    attribute pn { text }?,
+    ((artwork | dl | figure | ol | sourcecode | t | ul)+
+     | (text
xref =
  element xref {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + attribute target { xsd:IDREF },
    - [ a:defaultValue = "false" ] attribute pageno { "true" | "false" }?,
    + [ a:defaultValue = "false" ]
    + attribute pageno { "true" | "false" }?,
    - attribute format { "counter" | "title" | "none" | "default" }?
    + attribute format { "default" | "title" | "counter" | "none" }
    + )?,
    + attribute derivedContent { text }?,
    + text
    + }
  + relref =
  + element relref {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + attribute target { xsd:IDREF },
    - attribute displayFormat { "of" | "comma" | "parens" | "bare" }?,
    + (attribute section { text },
    + attribute relative { text })?},
    + attribute derivedRemoteContent { text }?,
    + attribute derivedLink { text }?,
    + text
  }
  eref =
  element eref {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + attribute target { text },
    + text
iref =
  element iref {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    attribute item { text },
    [ a:defaultValue = "" ] attribute subitem { text }?,
    [ a:defaultValue = "false" ]
    attribute primary { "true" | "false" }?,
    + [ a:defaultValue = "" ] attribute irefid { text }?,
    empty
  }

cref =
  element cref {
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    attribute anchor { xsd:ID }?,
    attribute source { text }?,
    text
    + [ a:defaultValue = "true" ]
    attribute display { "true" | "false" }?,
    + (text | em | eref | relref | strong | sub | sup | tt | xref)*
    + }
    + tt =
    + element tt {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      + (text
      +    | bcp14
      +    | cref
      +    | em
      +    | eref
      +    | iref
      +    | relref
      +    | strong
      +    | sub
      +    | sup
      +    | xref)*
      + }
    + strong =
    + element strong {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      + (text
      +    | bcp14
      +    | cref
      +    | em
      +    | eref
+  | iref
+  | relref
+  | sub
+  | sup
+  | tt
+  | xref)*
+ }
+ em =
+  element em {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
+    (text
+    +  bcp14
+    +  cref
+    +  eref
+    +  iref
+    +  relref
+    +  strong
+    +  sub
+    +  sup
+    +  tt
+    +  xref)*
+  }
+ sub =
+  element sub {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
+    (text
+    +  bcp14
+    +  cref
+    +  em
+    +  eref
+    +  iref
+    +  relref
+    +  strong
+    +  tt
+    +  xref)*
+  }
+ sup =
+  element sup {
+    attribute xml:base { text }?,
+    attribute xml:lang { text }?,
+    (text
+    +  bcp14
+    +  cref
+    +  em
+    +  eref
+    +  iref
+ | relref
+ | strong
+ | tt
+ | xref)*
}

spanx =
element spanx {
+ attribute xml:base { text }?,
+ attribute xml:lang { text }?,
[ a:defaultValue = "preserve" ]
attribute xml:space { "default" | "preserve" }?,
[ a:defaultValue = "emph" ] attribute style { text }?,
text
}

vspace =
element vspace {
+ attribute xml:base { text }?,
+ attribute xml:lang { text }?,
[ a:defaultValue = "0" ] attribute blankLines { text }?,
empty
}

figure =
element figure {
+ attribute xml:base { text }?,
+ attribute xml:lang { text }?,
attribute anchor { xsd:ID }?,
+ attribute pn { text }?,
[ 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 }?,
+ name?,
+ iref*,
preamble?,
- artwork,
+ (artwork | sourcecode)+,
postamble?
}

+ table =
element table {
+ attribute xml:base { text }?,
+ attribute xml:lang { text }?,
+ attribute anchor { xsd:ID }?,

Hoffman                   Expires March 7, 2016               [Page 131]
preamble =
  element preamble { (text | xref | eref | iref | cref | spanx)* }
+ element preamble {
  + attribute xml:base { text }?,
  + attribute xml:lang { text }?,
  + (text
  +   | bcp14
  +   | cref
  +   | em
  +   | iref
  +   | relref
  +   | spanx
  +   | strong
  +   | sup
  +   | tt
  +   | xref)*
  + }
artwork =
  element artwork {
    + attribute xml:space { "default" | "preserve" }?,
    + attribute xml:base { text }?,
    + attribute xml:lang { text }?,
    + attribute anchor { xsd:ID }?,
    + attribute pn { text }?,
    + attribute xml:space { text }?,
    [ a:defaultValue = "left" ]
    + attribute align { "left" | "center" | "right" }?,
    [ a:defaultValue = "" ]
    + attribute alt { text }?,
    [ a:defaultValue = "" ]
    + attribute width { text }?,
    [ a:defaultValue = "" ]
    + attribute height { text }?,
    + text*
    + attribute originalSrc { text }?,
    + (text* | svg)
    + svg =
element ns1:svg {
  [ a:defaultValue = "can be found in draft-brownlee-svg-rfc" ]
  attribute TheRealDefinition { text }*
  }

sourcecode =
  element sourcecode {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  [ a:defaultValue = "" ] attribute name { text }?,
  [ a:defaultValue = "" ] attribute type { text }?,
  attribute src { text }?,
  attribute originalSrc { text }?,
  text
  }

thead =
  element thead {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  tr+
  }

tbody =
  element tbody {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  tr+
  }

tfoot =
  element tfoot {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  tr+
  }

t =
  element tr {
  attribute xml:base { text }?,
  attribute xml:lang { text }?,
  attribute anchor { xsd:ID }?,
  attribute pn { text }?,
  [ a:defaultValue = "0" ] attribute border { text }?,
  (td | th)+
+ )
+ td =
+   element td {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute anchor { xsd:ID }?,
+     [ a:defaultValue = "0" ] attribute border { text }?,
+     [ a:defaultValue = "0" ] attribute colspan { text }?,
+     [ a:defaultValue = "0" ] attribute rowspan { text }?,
+     [ a:defaultValue = "left" ]
+     attribute align { "left" | "center" | "right" }?,
+     t+
+     | (text
+     | artwork
+     | bcp14
+     | br
+     | cref
+     | dl
+     | em
+     | eref
+     | figure
+     | iref
+     | ol
+     | relref
+     | sourcecode
+     | strong
+     | sub
+     | sup
+     | tt
+     | ul
+     | xref)+
+   }
+ th =
+   element th {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     attribute anchor { xsd:ID }?,
+     attribute pn { text }?,
+     [ a:defaultValue = "0" ] attribute border { text }?,
+     [ a:defaultValue = "0" ] attribute colspan { text }?,
+     [ a:defaultValue = "0" ] attribute rowspan { text }?,
+     [ a:defaultValue = "left" ]
+     attribute align { "left" | "center" | "right" }?,
+     t+
+     | (text
+     | artwork
+     | bcp14
+     | br
postamble =
  element postamble { (text | xref | eref | iref | cref | spanx)*
    + element postamble {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      + (text | cref | eref | iref | spanx | xref)*
    }
  texttable =
    element texttable {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      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" }?,
      name?,
      preamble?,
      ttcol+,
      c*,
      postamble?
    }
  ttcol =
    element ttcol {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      attribute width { text }?,
      [ a:defaultValue = "left" ]
      attribute align { "left" | "center" | "right" }?,
  }
Hoffman                   Expires March 7, 2016               [Page 135]
+ (cref | eref | iref | xref | text)*
+ }
+ c =
+ element c {
+   attribute xml:base { text }?,
+   attribute xml:lang { text }?,
+   (text | cref | eref | iref | spanx | xref)*
+ }
+ bcp14 =
+ element bcp14 {
+   attribute xml:base { text }?,
+   attribute xml:lang { text }?,
+   text
+ }
- c = element c { (text | xref | eref | iref | cref | spanx)* }  
- back = element back { references*, section* }
+ br =
+ element br {
+   attribute xml:base { text }?,
+   attribute xml:lang { text }?,
+   empty
+ }
+ back =
+ element back {
+   attribute xml:base { text }?,
+   attribute xml:lang { text }?,
+   displayreference*,
+   references*,
+   section*
+ }
+ displayreference =
+ element displayreference {
+   attribute xml:base { text }?,
+   attribute xml:lang { text }?,
+   attribute target { xsd:IDREF },
+   attribute to { text }
+ }
+ references =
+ element references {
+   [ a:defaultValue = "References" ] attribute title { text }?,
+   reference+
+   attribute xml:base { text }?,
+   attribute xml:lang { text }?,
+   attribute anchor { xsd:ID }?,
+   attribute title { text }?,
+   name?,
+   (reference | referencegroup)*
+ }
reference =
   element reference {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      attribute anchor { xsd:ID },
      attribute target { text }?,
      + [ a:defaultValue = "true" ]
      + attribute quoteTitle { "true" | "false" }?,
      front,
      - seriesInfo*,
      - format*,
      - annotation*
      + (annotation | format | refcontent | seriesInfo)*
   } +
referencegroup =
   element referencegroup {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      attribute anchor { xsd:ID },
      + reference+
   }
seriesInfo =
   element seriesInfo {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      attribute name { text },
      attribute value { text },
      + attribute asciiName { text }?,
      + attribute asciiValue { text }?,
      empty
   }
format =
   element format {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      attribute target { text }?,
      attribute type { text },
      attribute octets { text }?,
      empty
   }
annotation =
   element annotation { (text | xref | eref | iref | cref |
      - spanx)* } +
   element annotation {
      + attribute xml:base { text }?,
      + attribute xml:lang { text }?,
      + (text
      + | bcp14
+       cref
+       em
+       eref
+       iref
+       relref
+       spanx
+       strong
+       sub
+       sup
+       tt
+       xref)*
+   }
+ refcontent =
+   element refcontent {
+     attribute xml:base { text }?,
+     attribute xml:lang { text }?,
+     (text | bcp14 | em | strong | sub | sup | tt)*
+   }
start = rfc

Index

A
abbrev attribute
   in organization element  40
   in title element  69
abstract element  11, 97
   anchor attribute  11
   inside front  33
address element  11, 97
   inside author  17
align attribute
   in artwork element  14
   in figure element  31
   in td element  66
   in texttable element  79
   in th element  68
   in ttcol element  81
alt attribute
   in artwork element  14
   in figure element  31
anchor attribute
   in abstract element  11
   in artwork element  15
   in aside element  17
   in blockquote element  20
   in cref element  23
in dd element 25
in dl element 27
in dt element 28
in figure element 31
in li element 36
in ol element 39
in reference element 43
in referencegroup element 44
in references element 45
in section element 54
in sourcecode element 58
in t element 63
in table element 64
in tbody element 64
in td element 66
in texttable element 80
in tfoot element 67
in th element 68
in thead element 69
in tr element 70
in ul element 71
annotation element 12, 97
   inside reference 43
application/rfc+xml Media Type 82
area element 12, 97
   inside front 33
artwork element 13, 97
   align attribute 14
   alt attribute 14
   anchor attribute 15
   height attribute 15
   inside aside 16
   inside blockquote 19
   inside dd 24
   inside figure 31
   inside li 35
   inside section 53
   inside td 65
   inside th 67
   name attribute 15
   src attribute 15
   type attribute 15
   width attribute 16
   xml:space attribute 16
ascii attribute
   in city element 21
   in code element 21
   in country element 22
in email element 29
in organization element 41
in postallLine element 42
in region element 45
in street element 59
in title element 69
asciiFullname attribute
in author element 17
asciiInitials attribute
in author element 17
asciiName attribute
in seriesInfo element 55
asciiSurname attribute
in author element 18
asciiValue attribute
in seriesInfo element 56
aside element 16, 97
anchor attribute 17
inside section 53
Attributes
abbrev 40, 69
align 14, 31, 66, 68, 79, 81
alt 14, 31
anchor 11, 15, 17, 20, 23, 25, 27–28, 31, 36, 39, 43–45, 54,
58, 63–64, 66–71, 80
ascii 21–22, 29, 41–42, 45, 59, 69
asciiFullname 17
asciiInitials 17
asciiName 55
asciiSurname 18
asciiValue 56
blankLines 81
border 66, 68, 70
category 49
cite 20
colspace 66, 69
consensus 49
counter 77
day 24
display 23
displayFormat 46
docName 50
empty 71
format 73
fullname 18
group 39
hangIndent 77
hanging 27
value 56
version 53
width 16, 32, 81
xml:space 16, 79
year 24
author element 17, 97
asciiFullname attribute 17
asciiInitials attribute 17
asciiSurname attribute 18
fullname attribute 18
initials attribute 18
inside front 33
role attribute 18
surname attribute 18

B
back element 18, 97
inside rfc 49
bcp14 element 19, 97
inside annotation 12
inside blockquote 20
inside c 75
inside dd 25
inside dt 27
inside em 28
inside li 35
inside postamble 77
inside preamble 78
inside refcontent 42
inside strong 60
inside sub 61
inside sup 61
inside t 62
inside td 65
inside th 67
inside tt 70
blankLines attribute
in vspace element 81
blockquote element 19, 97
anchor attribute 20
cite attribute 20
inside section 53
quotedFrom attribute 20
boilerplate element 21, 97
inside front 33
border attribute
in td element 66
in th element 68
in tr element 70
br element 21, 97
inside td 65
inside th 67

c

c element 75, 97
inside texttable 79
category attribute
in rfc element 49
cite attribute
in blockquote element 20
city element 21, 97
ascii attribute 21
inside postal 41
code element 21, 97
ascii attribute 21
inside postal 41
colspan attribute
in td element 66
in th element 69
consensus attribute
in rfc element 49
counter attribute
in list element 77
country element 22, 97
ascii attribute 22
inside postal 41
cref element 22, 97
anchor attribute 23
display attribute 23
inside annotation 12
inside blockquote 20
inside c 75
inside dd 25
inside dt 28
inside em 29
inside li 35
inside name 37
inside postamble 77
inside preamble 78
inside strong 60
inside sub 61
inside sup 61
inside t 62
inside td 65
inside th 67
inside tt 71
inside ttcol 80
source attribute 23

D
date element 23, 97
day attribute 24
inside front 33
month attribute 24
year attribute 24
day attribute
in date element 24
dd element 24, 97
anchor attribute 25
inside dl 27
display attribute
in cref element 23
displayFormat attribute
in relref element 46
displayreference element 26, 97
inside back 18
target attribute 26
to attribute 26
dl element 26, 97
anchor attribute 27
hanging attribute 27
inside abstract 11
inside aside 16
inside blockquote 19
inside dd 25
inside li 35
inside note 38
inside section 53
inside td 65
inside th 67
spacing attribute 27
docName attribute
in rfc element 50
dt element 27, 97
anchor attribute 28
inside dl 27

E
Elements
abstract 11, 33
address 11, 17
annotation 12, 43
area 12, 33
artwork 13, 16, 19, 24, 31, 35, 53, 65, 67
inside tt  71
email element  29, 97
    ascii attribute  29
    inside address  11
empty attribute
    in ul element  71
deref element  29, 97
    inside annotation  12
    inside blockquote  20
    inside c  75
    inside cref  22
    inside dd  25
    inside dt  28
    inside em  29
    inside li  35
    inside name  37
    inside postamble  77
    inside preamble  78
    inside strong  60
    inside sub  61
    inside sup  61
    inside t  62
    inside td  65
    inside th  68
    inside tt  71
    inside ttcol  80
    target attribute  30

facsimile element  76, 97
    inside address  11
figure element  30, 97
    align attribute  31
    alt attribute  31
    anchor attribute  31
    height attribute  32
    inside aside  16
    inside blockquote  19
    inside dd  25
    inside li  35
    inside section  53
    inside td  65
    inside th  68
    src attribute  32
    suppress-title attribute  32
    title attribute  32
    width attribute  32
    format attribute
in xref element  73
format element  76, 97
   inside reference  43
octets attribute  76
target attribute  76
type attribute  76
front element  32, 97
   inside reference  43
   inside rfc  49
fullname attribute
   in author element  18

G
   group attribute
      in ol element  39

H
   hangIndent attribute
      in list element  77
   hanging attribute
      in dl element  27
   hangText attribute
      in t element  63
height attribute
   in artwork element  15
   in figure element  32
href attribute
   in link element  36

I
   indexInclude attribute
      in rfc element  50
   initials attribute
      in author element  18
   ipr attribute
      ‘*2026’  90
      ‘*3667’  90
      ‘*3978’  90
      ‘*trust200811’  90
      ‘*trust200902’  88
      ‘noDerivativesTrust200902’  89
      ‘noModificationTrust200902’  89
      ‘pre5378Trust200902’  89
      ‘trust200902’  89
      in rfc element  50
   iprExtract attribute
      in rfc element  51
   iref element  33, 97
inside annotation 12
inside aside 16
inside blockquote 20
inside c 75
inside dd 25
inside dt 28
inside em 29
inside figure 31
inside li 35
inside postamble 77
inside preamble 78
inside section 53
inside strong 60
inside sub 61
inside sup 62
inside t 62
inside table 64
inside td 65
inside th 68
inside tt 71
inside ttcol 80
item attribute 34
primary attribute 34
subitem attribute 34
item attribute
in iref element 34

K
keepWithNext attribute
in t element 63
keepWithPrevious attribute
in t element 63
keyword element 34, 97
inside front 33

L
li element 34, 97
anchor attribute 36
inside ol 38
inside ul 71
link element 36, 97
href attribute 36
inside rfc 49
rel attribute 36
list element 76, 97
counter attribute 77
hangIndent attribute 77
inside aside 16
inside t 62
style attribute 77

M
Media Type
application/rfc+xml 82
middle element 37, 97
inside rfc 49
month attribute
in date element 24

N
name attribute
in artwork element 15
in seriesInfo element 56
in sourcecode element 58
name element 37, 97
inside figure 31
inside note 38
inside references 45
inside section 53
inside table 64
inside texttable 79
note element 37, 97
inside front 33
removeInRFC attribute 38
title attribute 38
number attribute
in rfc element 51
numbered attribute
in section element 54

O
obsoletes attribute
in rfc element 51
octets attribute
in format element 76
ol element 38, 97
anchor attribute 39
group attribute 39
inside abstract 11
inside aside 16
inside blockquote 19
inside dd 25
inside li 35
inside note 38
inside section 53
inside td 65
inside th 68
spacing attribute 39
start attribute 39
type attribute 39
organization element 40, 97
abbrev attribute 40
ascii attribute 41
inside author 17

P
pageno attribute
  in xref element 74
phone element 41, 97
  inside address 11
postal element 41, 97
  inside address 11
postalLine element 42, 97
  ascii attribute 42
  inside postal 42
postamble element 77, 97
  inside figure 31
  inside texttable 79
preamble element 78, 97
  inside figure 31
  inside texttable 79
prepTime attribute
  in rfc element 51
primary attribute
  in iref element 34

Q
quotedFrom attribute
  in blockquote element 20
quoteTitle attribute
  in reference element 43

R
refcontent element 42, 97
  inside reference 43
reference element 43, 97
  anchor attribute 43
  inside referencegroup 44
  inside references 45
  quoteTitle attribute 43
target attribute 44
referencegroup element 44, 97
  anchor attribute 44
  inside references 45
references element 44, 97
  anchor attribute 45
  inside back 18
  title attribute 45
region element 45, 97
  ascii attribute 45
  inside postal 41
rel attribute
  in link element 36
relative attribute
  in relref element 48
relref element 45, 97
  displayFormat attribute 46
  inside annotation 12
  inside blockquote 20
  inside cref 22
  inside dd 25
  inside dt 28
  inside em 29
  inside li 35
  inside name 37
  inside strong 60
  inside sup 61
  inside sup 62
  inside t 62
  inside td 65
  inside th 68
  inside tt 71
relative attribute 48
section attribute 49
target attribute 49
removeInRFC attribute
  in note element 38
  in section element 54
category attribute 49
consensus attribute 49
docName attribute 50
indexInclude attribute 50
ipr attribute 50
iprExtract attribute 51
number attribute 51
obsoletes attribute 51
prepTime attribute 51
seriesNo attribute 51
sortRefs attribute 51
submissionType attribute 52
symRefs attribute 52
tocDepth attribute 52
tocInclude attribute 52
updates attribute 53
version attribute 53
role attribute
  in author element 18
rowspan attribute
  in td element 66
  in th element 69

section attribute
  in relref element 49
section element 53, 97
  anchor attribute 54
  inside back 18
  inside boilerplate 21
  inside middle 37
  inside section 54
  numbered attribute 54
removeInRFC attribute 54
title attribute 54
toc attribute 55
seriesInfo element 55, 97
  asciiName attribute 55
  asciiValue attribute 56
  inside front 33
  inside reference 43
  name attribute 56
  value attribute 56
seriesNo attribute
  in rfc element 51
sortRefs attribute
  in rfc element 51
source attribute
  in cref element 23
sourcecode element 57, 97
  anchor attribute 58
  inside blockquote 19
  inside dd 25
  inside figure 31
  inside li 35
  inside section 54
  inside td 65
  inside th 68
  name attribute 58
src attribute 58
type attribute 58
spacing attribute
  in dl element 27
  in ol element 39
  in ul element 72
spanx element 78, 97
  inside annotation 12
  inside c 75
  inside postamble 77
  inside preamble 78
  inside t 62
  style attribute 79
xml:space attribute 79
src attribute
  in artwork element 15
  in figure element 32
  in sourcecode element 58
start attribute
  in ol element 39
street element 59, 97
  ascii attribute 59
  inside postal 41
strong element 59, 97
  inside annotation 12
  inside blockquote 20
  inside c 75
  inside cref 22
  inside dd 25
  inside dt 28
  inside em 29
  inside li 35
  inside postamble 77
  inside preamble 78
  inside refcontent 43
  inside sub 61
  inside sup 62
  inside t 62
  inside td 65
  inside th 68
  inside tt 71
style attribute
  in list element 77
  in spanx element 79
  in texttable element 80
sub element 60, 97
  inside annotation 12
  inside blockquote 20
  inside c 76
  inside cref 22
inside dd  25
inside dt  28
inside em  29
inside li  35
inside postamble  77
inside preamble  78
inside refcontent  43
inside strong  60
inside t  62
inside td  65
inside th  68
inside tt  71
subitem attribute
  in iref element  34
submissionType attribute
  in rfc element  52
sup element  61, 97
  inside annotation  12
  inside blockquote  20
  inside c  76
  inside cref  22
  inside dd  25
  inside dt  28
  inside em  29
  inside li  35
  inside postamble  77
  inside preamble  78
  inside refcontent  43
  inside strong  60
  inside t  62
  inside td  66
  inside th  68
  inside tt  71
suppress-title attribute
  in figure element  32
  in texttable element  80
surname attribute
  in author element  18
svg element  75, 97
  inside artwork  14
  TheRealDefinition attribute  75
symRefs attribute
  in rfc element  52

T
t element  62, 97
  anchor attribute  63
  hangText attribute  63
inside abstract 11
inside aside 16
inside blockquote 19
inside dd 25
inside li 35
inside list 77
inside note 38
inside section 54
inside td 65
inside th 67
keepWithNext attribute 63
keepWithPrevious attribute 63
table element 64, 97
    anchor attribute 64
    inside aside 17
    inside section 54
target attribute
    in displayreference element 26
    in eref element 30
    in format element 76
    in reference element 44
    in relref element 49
    in xref element 74
tbody element 64, 97
    anchor attribute 64
    inside table 64
td element 65, 97
    align attribute 66
    anchor attribute 66
    border attribute 66
    colspan attribute 66
    inside tr 70
    rowspan attribute 66
texttable element 79, 97
    align attribute 79
    anchor attribute 80
    inside section 54
    style attribute 80
    suppress-title attribute 80
    title attribute 80
tfoot element 67, 97
    anchor attribute 67
    inside table 64
th element 67, 97
    align attribute 68
    anchor attribute 68
    border attribute 68
    colspan attribute 69
inside tr 70
rowspan attribute 69
thead element 69, 97
anchor attribute 69
inside table 64
TheRealDefinition attribute
in svg element 75
title attribute
in figure element 32
in note element 38
in references element 45
in section element 54
in texttable element 80
title element 69, 97
abbrev attribute 69
ascii attribute 69
inside front 33
to attribute
in displayreference element 26
toc attribute
in section element 55
tocDepth attribute
in rfc element 52
tocInclude attribute
in rfc element 52
t element 70, 97
anchor attribute 70
border attribute 70
inside tbody 64
inside tfoot 67
inside thead 69
tt element 70, 97
inside annotation 12
inside blockquote 20
inside c 76
inside cref 23
inside dd 25
inside dt 28
inside em 29
inside li 35
inside name 37
inside postamble 78
inside preamble 78
inside refcontent 43
inside strong 60
inside sub 61
inside sup 62
inside t 63
inside td 66
inside th 68
ttcol element 80, 97
  align attribute 81
  inside texttable 79
  width attribute 81
type attribute
  in artwork element 15
  in format element 76
  in ol element 39
  in sourcecode element 58

ul element 71, 97
  anchor attribute 71
  empty attribute 71
  inside abstract 11
  inside aside 17
  inside blockquote 19
  inside dd 25
  inside li 35
  inside note 38
  inside section 54
  inside td 66
  inside th 68
  spacing attribute 72
updates attribute
  in rfc element 53
uri element 72, 97
  inside address 11

value attribute
  in seriesInfo element 56
version attribute
  in rfc element 53
vspace element 81, 97
  blankLines attribute 81
  inside t 63

width attribute
  in artwork element 16
  in figure element 32
  in ttcol element 81
workgroup element 72, 97
  inside front 33
X

xml:space attribute
  in artwork element  16
  in spanx element  79
xref element  72, 97
  format attribute  73
  inside annotation  12
  inside blockquote  20
  inside c  76
  inside cref  23
  inside dd  25
  inside dt  28
  inside em  29
  inside li  35
  inside name  37
  inside postamble  78
  inside preamble  78
  inside strong  60
  inside sub  61
  inside sup  62
  inside t  63
  inside td  66
  inside th  68
  inside tt  71
  inside ttcol  80
  pageno attribute  74
  target attribute  74
xref formats
  counter  73
  default  73
  none  74
  title  74

Y

year attribute
  in date element  24

Author’s Address

Paul Hoffman
ICANN

EMail: paul.hoffman@icann.org
Abstract

This document defines the ’XML2RFC’ version 2 vocabulary; an XML-based language used for writing RFCs and Internet-Drafts.

Version 2 represents the current state of the vocabulary (as implemented by several tools and as used by the RFC Editor) around 2014.

Editorial Note (To be removed by RFC Editor)

Discussion of this draft takes place on the XML2RFC mailing list (xml2rfc@ietf.org), which has its home page at <https://www.ietf.org/mailman/listinfo/xml2rfc>.

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 July 11, 2015.

Copyright Notice

Copyright (c) 2015 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.

Table of Contents

1.  Introduction ..................................................  4
   1.1.  Syntax Notation ..........................................  4
2.  Elements .......................................................  4
   2.1.  <abstract> .................................................  5
   2.2.  <address> .................................................  5
   2.3.  <annotation> ..............................................  5
   2.4.  <area> .....................................................  6
   2.5.  <artwork> ................................................  6
   2.6.  <author> ..................................................  9
   2.7.  <back> ..................................................... 10
   2.8.  <c> ....................................................... 10
   2.9.  <city> ..................................................... 11
   2.10. <code> .................................................... 11
   2.11. <country> ................................................ 11
   2.12. <cref> .................................................... 11
   2.13. <date> .................................................... 12
   2.14. <email> ................................................... 13
   2.15. <eref> .................................................... 13
   2.16. <facsimile> .............................................. 13
   2.17. <figure> .................................................. 14
   2.18. <format> .................................................. 15
   2.19. <front> ................................................... 16
   2.20. <iref> ..................................................... 16
   2.21. <keyword> ................................................ 17
   2.22. <list> ..................................................... 18
   2.23. <middle> .................................................. 20
   2.24. <note> .................................................... 20
   2.25. <organization> ........................................... 20
   2.26. <phone> ................................................... 21
   2.27. <postal> .................................................. 21
   2.28. <postamble> .............................................. 22
   2.29. <preamble> ............................................... 22
   2.30. <reference> .............................................. 23
   2.31. <references> ............................................. 23
   2.32. <region> .................................................. 24
   2.33. <rfc> ...................................................... 24
   2.34. <section> ................................................ 28
   2.35. <seriesInfo> ............................................. 29
   2.36. <spanx> ................................................... 29
   2.37. <street> ................................................ 30
1. Introduction

This document describes version 2 ("v2") of the ‘XML2RFC’ vocabulary; an XML-based language (‘Extensible Markup Language’, [XML]) used for writing RFCs ([RFC7322]) and Internet-Drafts ([IDGUIDE]).

Version 2 represents the current state of the vocabulary (as implemented by several tools and as used by the RFC Editor) around 2014.

It obsoletes the original version ("v1") [RFC2629], which contained the original language definition, and which was subsequently extended. Many of the changes leading to version 2 have been described in "Writing I-Ds and RFCs using XML (revised)" ([V1rev]), but that document has not been updated since 2008.

Processing Instructions (Section 2.6 of [XML]) generally are specific to a given processor, and thus are not considered to be part of the vocabulary. See Section 4.1 of [TCLReadme] for a list of the processing instructions supported by the first implementation of an xml2rfc processor.

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

1.1. 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.

Except inside <artwork>, horizontal whitespace and linebreaks are collapsed into a single whitespace, and leading and trailing whitespace are trimmed off.
2.1. <abstract>

Contains the abstract of the document. The abstract ought to be self-contained and thus should not contain references or unexpanded abbreviations. See Section 4.3 of [RFC7322] for more information.

This element appears as a child element of: <front> (Section 2.19).

Content model:

One or more <t> elements (Section 2.38)

2.2. <address>

Provides address information for the author.

This element appears as a child element of: <author> (Section 2.6).

Content model:

In this order:
1. One optional <postal> element (Section 2.27)
2. One optional <phone> element (Section 2.26)
3. One optional <facsimile> element (Section 2.16)
4. One optional <email> element (Section 2.14)
5. One optional <uri> element (Section 2.42)

2.3. <annotation>

Provides additional prose augmenting a bibliographical reference.

This element appears as a child element of: <reference> (Section 2.30).

Content model:

In any order:

- Text

- <xref> elements (Section 2.45)
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 full name or the abbreviation of one of the IETF areas as listed on <http://www.ietf.org/iesg/area.html>. The list at the time that this document is being published is: "Applications", "app", "General", "gen", "Internet", "int", "Operations and Management", "ops", "Real-time Applications and Infrastructure", "rai", "Routing", "rtg", "Security", "sec", "Transport", "tsv".

This element appears as a child element of: <front> (Section 2.19).

Content model: only text content.

2.5.  <artwork>

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

<artwork> is the only element in the vocabulary that provides full control of horizontal whitespace and line breaks, and thus is used for a variety of things, such as:

- diagrams ("line art"),
- source code,
- formal languages (such as ABNF or the RNC notation used in this document),
- message flow diagrams,
- complex tables, or
- protocol unit diagrams.

Note that processors differ in the handling of horizontal TAB characters (some expand them, some treat them as single space) and
thus these ought to be avoided.

Alternatively, the "src" attribute allows referencing an external graphics file, such as a bitmap or a vector drawing, using a URI ("Uniform Resource Identifier", [RFC3986]). 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. Note that RFCs occasionally are published with enhanced diagrams; a recent example is [RFC5598].

This element appears as a child element of: <figure> (Section 2.17).

Content model:

Text

2.5.1. 'align' attribute

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

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

The suggested height of the graphics included using the "src" attribute.

This attribute is format-dependent and ought to be avoided.

When generating HTML output, current implementations copy the attribute "as is". For other output formats it is usually ignored.

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 source code).

2.5.5. ‘src’ attribute

The URI reference of a graphics file ([RFC3986], Section 4.1).

Note that this can be a "data" URI ([RFC2397]) as well; in which case the graphics file is wholly part of the XML file.

2.5.6. ‘type’ attribute

Specifies the type of the artwork.

The value either is an Internet Media Type (see [RFC2046]) or a keyword (such as "abnf"). The set of recognized keywords varies across implementations.

How it is used depends on context and application. For instance, a formatter can attempt to syntax-highlight code in certain known languages.

2.5.7. ‘width’ attribute

The suggested width of the graphics included using the "src" attribute.

This attribute is format-dependent and ought to be avoided.

When generating HTML output, current implementations copy the attribute "as is". For other output formats it is usually ignored.

2.5.8. ‘xml:space’ attribute

Determines whitespace handling.

"preserve" is both the default value and the only meaningful setting anyway (because that’s what the <artwork> element is for).

See also Section 2.10 of [XML].

Allowed values:

- "default"
- "preserve" (default)
2.6. <author>

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

The <author> elements contained within the document’s <front> element are used to fill the boilerplate, and also to generate the "Author’s Address" section (see Section 4.12 of [RFC7322]).

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

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

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

This element appears as a child element of: <front> (Section 2.19).

Content model:

In this order:

1. One optional <organization> element (Section 2.25)
2. One optional <address> element (Section 2.2)

2.6.1. ‘fullname’ attribute

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

2.6.2. ‘initials’ attribute

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

Initials should be provided as a whitespace separated list of pairs of a letter and a dot.

2.6.3. ‘role’ attribute

Specifies the role the author had in creating the document.
Allowed values:
  - "editor"

2.6.4. ‘surname’ attribute

The author’s surname (used on the front page and in references).

2.7. <back>

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

This element appears as a child element of: <rfc> (Section 2.33).

Content model:

In this order:
1. Optional <references> elements (Section 2.31)
2. Optional <section> elements (Section 2.34)

2.8. <c>

Provides the content of a cell in a table.

This element appears as a child element of: <texttable> (Section 2.39).

Content model:

In any order:
  - Text
  - <xref> elements (Section 2.45)
  - <eref> elements (Section 2.15)
  - <iref> elements (Section 2.20)
  - <cref> elements (Section 2.12)
  - <spanx> elements (Section 2.36)
2.9.  <city>

Gives the city name in a postal address.

This element appears as a child element of: <postal> (Section 2.27).

Content model: only text content.

2.10.  <code>

Gives the postal region code.

This element appears as a child element of: <postal> (Section 2.27).

Content model: only text content.

2.11.  <country>

Gives the country in a postal address.

This element appears as a child element of: <postal> (Section 2.27).

Content model: only text content.

2.12.  <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 a child element of: <annotation> (Section 2.3), <c> (Section 2.8), <postamble> (Section 2.28), <preamble> (Section 2.29), and <t> (Section 2.38).

Content model: only text content.

2.12.1.  ‘anchor’ attribute

Document-wide unique identifier for this comment. The processor will auto-generate an identifier when none is given.

The value needs to be a valid XML "Name" (Section 2.3 of [XML]), additionally constrained to US-ASCII characters ([USASCII]).
2.12.2. ‘source’ attribute

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

2.13. <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.

In the boilerplate case, it defines the publication date, which, when producing Internet-Drafts, will be used for computing the expiration date (see Section 8 of [IDGUIDE]). When one or more of "year", "month", or "day" are left out, the processor will attempt to use the current system date if the attributes that are present are consistent with that date.

Note that in this case, month names need to match the full (English) month name ("January", "February", "March", "April", "May", "June", "July", "August", "September", "October", "November", or "December") in order for expiration calculations to work (some implementations might support additional formats, though).

In the case of bibliographic references, the date information can have prose text for the month or year. For example, vague dates (year="ca. 2000"), date ranges (year="2012-2013"), non-specific months (month="Second quarter") and so on, are allowed.

This element appears as a child element of: <front> (Section 2.19).

Content model: this element does not have any contents.

2.13.1. ‘day’ attribute

In the "boilerplate" case: the day of publication; this is a number. Otherwise: an indication of the publication day, with the format not being restricted.

2.13.2. ‘month’ attribute

In the "boilerplate" case: the month of publication; this is the English name of the month. Otherwise: an indication of the publication month, with the format not being restricted.
2.13.3. ‘year’ attribute

In the "boilerplate" case: the year of publication; this is a number (usually four-digit). Otherwise: an indication of the publication year, with the format not being restricted.

2.14. <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 a child element of: <address> (Section 2.2).

Content model: only text content.

2.15. <eref>

Represents an "external" link (as specified in the "target" attribute).

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).

[oi-eref: Need to discuss what we can say for non-empty erefs (also whether requesting angle brackets is ok for all output formats)]

This element appears as a child element of: <annotation> (Section 2.3), <c> (Section 2.8), <postamble> (Section 2.28), <preamble> (Section 2.29), and <t> (Section 2.38).

Content model: only text content.

2.15.1. ‘target’ attribute (mandatory)

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

2.16. <facsimile>

Represents the phone number of a fax machine.

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 a child element of: <address> (Section 2.2).

Content model: only text content.

2.17. <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 a child element of: <section> (Section 2.34), and <t> (Section 2.38).

Content model:

In this order:
1. Optional <iref> elements (Section 2.20)
2. One optional <preamble> element (Section 2.29)
3. One <artwork> element (Section 2.5)
4. One optional <postamble> element (Section 2.28)

2.17.1. ‘align’ attribute

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

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

Allowed values:
- "left" (default)
- "center"
- "right"

2.17.2. ‘alt’ attribute

Duplicates functionality available on <artwork>; avoid it.

2.17.3. ‘anchor’ attribute

Document-wide unique identifier for this figure.

Furthermore, the presence of this attribute causes the figure to be
numbered.
The value needs to be a valid XML "Name" (Section 2.3 of [XML]).

2.17.4. ‘height’ attribute
Duplicates functionality available on <artwork>; avoid it.

2.17.5. ‘src’ attribute
Duplicates functionality available on <artwork>; avoid it.

2.17.6. ‘suppress-title’ attribute
Figures that have an "anchor" attribute will automatically get an
generated title (such as "Figure 1"), even if the "title"
attribute is absent. Setting this attribute to "true" will prevent
this.

Allowed values:
- "true"
- "false" (default)

2.17.7. ‘title’ attribute
The title for the figure; this usually appears on a line after the
figure.

2.17.8. ‘width’ attribute
Duplicates functionality available on <artwork>; avoid it.

2.18. <format>
Provides a link to an additional format variant for a reference.

Note that these additional links are neither used in published RFCs,
nor supported by all tools. If the goal is to provide a single URI
for a reference, the "target" attribute on <reference> can be used
instead.

This element appears as a child element of: <reference>
(Section 2.30).

Content model: this element does not have any contents.
2.18.1. "octets" attribute

Octet length of linked-to document.

2.18.2. "target" attribute

URI of document.

2.18.3. "type" attribute (mandatory)

The type of the linked-to document, such as "TXT", "HTML", or "PDF".

2.19. <front>

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

This element appears as a child element of: <reference>
(Section 2.30), and <rfc> (Section 2.33).

Content model:

In this order:

1. One <title> element (Section 2.40)
2. One or more <author> elements (Section 2.6)
3. One <date> element (Section 2.13)
4. Optional <area> elements (Section 2.4)
5. Optional <workgroup> elements (Section 2.44)
6. Optional <keyword> elements (Section 2.21)
7. One optional <abstract> element (Section 2.1)
8. Optional <note> elements (Section 2.24)

2.20. <iref>

Provides terms for the document’s index.

Index entries can be either be regular entries (when just the "item"
attribute is given) or nested entries (by specifying "subitem" as
well), grouped under a regular entry.
In this document, for instance, every element definition appears as a regular index entry ("iref element 2.20"). In addition, for each use of that element inside another parent element, a nested entry was added ("iref element 2.20, ... inside annotation 2.3").

Index entries generally refer to the exact place where the <iref> element occurred. An exception is the occurrence as a child element of <section>, in which case the whole section is considered to be relevant for that index entry. In some formats, index entries of this type might be displayed as range.

This element appears as a child element of: <annotation> (Section 2.3), <c> (Section 2.8), <figure> (Section 2.17), <postamble> (Section 2.28), <preamble> (Section 2.29), <section> (Section 2.34), and <t> (Section 2.38).

Content model: this element does not have any contents.

2.20.1. ‘item’ attribute (mandatory)

The item to include.

2.20.2. ‘primary’ attribute

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

Allowed values:

- "true"
- "false" (default)

2.20.3. ‘subitem’ attribute

The subitem to include.

2.21. <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 documents.

This element appears as a child element of: <front> (Section 2.19).
2.22. <list>

Delineates a text list.

Each list item is represented by a <t> element. The vocabulary currently does not directly support list items consisting of multiple paragraphs; if this is needed, <vspace> (Section 2.43) can be used as a workaround.

This element appears as a child element of: <t> (Section 2.38).

Content model:
One or more <t> elements (Section 2.38)

2.22.1. ‘counter’ attribute

This attribute holds a token that serves as an identifier for a counter. The intended use is continuation of lists, where the counter will be incremented for every list item, and there is no way to reset the counter.

Note that this attribute functions only when the style attribute is using the "format..." syntax (Section 2.22.3); otherwise, it is ignored.

2.22.2. ‘hangIndent’ attribute

For list styles with potentially wide labels, this attribute can override the default indentation level, measured in number of characters.

Note that it only affects style with variable-width labels ("format..." and "hanging", see below), and it may not affect formats in which the list item text appears _below_ the label.

2.22.3. ‘style’ attribute

This attribute is used to control the display of a list.

The value of this attribute is inherited by any nested lists that do not have this attribute set. It may be set to:

"empty"
For unlabeled list items; it can also be used for indentation purposes (this is the default value when there is an enclosing list where the style is specified).

"hanging"

For lists where the items are labeled with a piece of text.

The label text is specified in the 'hangText' attribute of the <t> element (Section 2.38.2).

"letters"

For ordered lists using letters as labels (lowercase letters followed by a period; after "z", it rolls over to a two-letter format). For nested lists, processors usually flip between uppercase and lowercase.

"numbers"

For ordered lists using numbers as labels.

"symbols"

For unordered (bulleted) lists.

The style of the bullets is chosen automatically by the processor (some implementations allow overriding the default using a processing instruction).

And, finally:

"format ...

For lists with customized labels, consisting of fixed text and an item counter in various formats.

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, etc.)
2.23. `<middle>`

Represents the main content of the document.

This element appears as a child element of: `<rfc>` (Section 2.33).

Content model:

One or more `<section>` elements (Section 2.34)

2.24. `<note>`

Creating 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 a child element of: `<front>` (Section 2.19).

Content model:

One or more `<t>` elements (Section 2.38)

2.24.1. ‘title’ attribute (mandatory)

The title of the note.

2.25. `<organization>`

Specifies the affiliation of an author.

This information appears in both the "Author’s Address" section and on the front page (see [RFC7322], Section 4.1.1 for more
information). If the value is long, an abbreviated variant can be specified in the "abbrev" attribute.

This element appears as a child element of: <author> (Section 2.6).

Content model: only text content.

2.25.1. ‘abbrev’ attribute

Abbreviated variant.

2.26. <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 a child element of: <address> (Section 2.2).

Content model: only text content.

2.27. <postal>

Contains child elements providing postal information.

Note that at least one <street> element needs to be present; however formatters will handle empty values just fine.

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

Content model:

In this order:

1. One or more <street> elements (Section 2.37)

2. In any order:
   * <city> elements (Section 2.9)
   * <region> elements (Section 2.32)
   * <code> elements (Section 2.10)
   * <country> elements (Section 2.11)
2.28. <postamble>

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

This element appears as a child element of: <figure> (Section 2.17),
and <texttable> (Section 2.39).

Content model:

In any order:

- Text
- <xref> elements (Section 2.45)
- <eref> elements (Section 2.15)
- <iref> elements (Section 2.20)
- <cref> elements (Section 2.12)
- <spanx> elements (Section 2.36)

2.29. <preamble>

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

This element appears as a child element of: <figure> (Section 2.17),
and <texttable> (Section 2.39).

Content model:

In any order:

- Text
- <xref> elements (Section 2.45)
- <eref> elements (Section 2.15)
- <iref> elements (Section 2.20)
- <cref> elements (Section 2.12)
- <spanx> elements (Section 2.36)
2.30. <reference>

Represents a bibliographical reference.

This element appears as a child element of: <references> (Section 2.31).

Content model:

In this order:

1. One <front> element (Section 2.19)
2. Optional <seriesInfo> elements (Section 2.35)
3. Optional <format> elements (Section 2.18)
4. Optional <annotation> elements (Section 2.3)

2.30.1. '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.

The value needs to be a valid XML "Name" (Section 2.3 of [XML]), additionally constrained to US-ASCII characters ([USASCII]). Thus, the character repertoire consists of "A-Z", "a-z", "0-9", ",", ",", ",", and ":", where "0-9", ",", and "," are disallowed as start character.

2.30.2. '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).

2.31. <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 Section 4.8.6 of [RFC7322]). This vocabulary supports the split with the "title" attribute.
By default, the order of references is significant. Processors however can be instructed to sort them based on their anchor names.

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

Content model:

One or more <reference> elements (Section 2.30)

2.31.1. ‘title’ attribute

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

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

2.32. <region>

Provides the region name in a postal address.

This element appears as a child element of: <postal> (Section 2.27).

Content model: only text content.

2.33. <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 2.19)
2. One <middle> element (Section 2.23)
3. One optional <back> element (Section 2.7)
2.33.1. ‘category’ attribute

Document category (see Appendix A.1).

Allowed values:

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

2.33.2. ‘consensus’ attribute

Affects the generated boilerplate.

See [RFC5741] for more information.

Allowed values:

- "no"
- "yes"

2.33.3. ‘docName’ attribute

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

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

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 Section 7 of [IDGUIDE] for further information.
2.33.4. ‘ipr’ attribute

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

Allowed values:

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

2.33.5. ‘iprExtract’ attribute

Identifies a single section within the document (by its ‘anchor’ attribute) for which extraction "as-is" is explicitly allowed (this is only relevant for historic values of the "ipr" attribute).
2.33.6. ‘number’ attribute

The number of the RFC to be produced.

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

2.33.7. ‘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.

2.33.8. ‘seriesNo’ attribute

Number within a document series.

The document series is defined by the "category" attribute; "seriesNo" is only applicable to the values "info" ("FYI" series), "std" ("STD" series), and "bcp" ("BCP" series).

2.33.9. ‘submissionType’ attribute

The document stream.

See Section 2 of [RFC5741] for details.

Allowed values:
- "IETF" (default)
- "IAB"
- "IRTF"
- "independent"

2.33.10. ‘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.
2.33.11. ‘xml:lang’ attribute

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

See Section 2.12 of [XML] for more information.

2.34. <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.

This element appears as a child element of: <back> (Section 2.7),<middle> (Section 2.23), and <section> (Section 2.34).

Content model:
In this order:
1. In any order:
   * <t> elements (Section 2.38)
   * <figure> elements (Section 2.17)
   * <texttable> elements (Section 2.39)
   * <iref> elements (Section 2.20)
2. Optional <section> elements (Section 2.34)

2.34.1. ‘anchor’ attribute

Document-wide unique identifier for this section.

The value needs to be a valid XML "Name" (Section 2.3 of [XML]).

2.34.2. ‘title’ attribute (mandatory)

The title of the section.

2.34.3. ‘toc’ attribute

Determines whether the section is included in the Table Of Contents.

The processor usually has defaults for whether a Table Of Contents
will be produced at all, and sections of which maximal depth will be included (frequently: 3). "include" and "exclude" allow overriding the processor's default behavior for the element they are specified on (they do not affect either nested or parent elements).

Allowed values:
- "include"
- "exclude"
- "default" (default)

2.35. <seriesInfo>

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

This element appears as a child element of: <reference> (Section 2.30).

Content model: this element does not have any contents.

2.35.1. ‘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".

2.35.2. ‘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).

2.36. <spanx>

Wraps a piece of text, indicating special formatting styles.

When generating plain text, processors usually emulate font changes using characters such as ** and _.

The following styles are defined:
emph  Simple emphasis (this is the default).

strong  Strong emphasis.

verb  "Verbatim" text (usually displayed using a monospaced font face).

This element appears as a child element of: <annotation> (Section 2.3), <c> (Section 2.8), <postamble> (Section 2.28), <preamble> (Section 2.29), and <t> (Section 2.38).

Content model: only text content.

2.36.1.  ‘style’ attribute

The style to be used (defaults to "emph").

2.36.2.  ‘xml:space’ attribute

Determines whitespace handling.

According to the DTD, the default value is "preserve". Tests however show that it doesn’t have any effect on processing; thus this attribute will be removed in future versions of the vocabulary.

See also Section 2.10 of [XML].

Allowed values:

  o  "default"

  o  "preserve" (default)

2.37.  <street>

Provides a street address.

This element appears as a child element of: <postal> (Section 2.27).

Content model: only text content.

2.38.  <t>

Contains a paragraph of text.

This element appears as a child element of: <abstract> (Section 2.1), <list> (Section 2.22), <note> (Section 2.24), and <section> (Section 2.34).
Content model:

In any order:

- Text
- <list> elements (Section 2.22)
- <figure> elements (Section 2.17)
- <xref> elements (Section 2.45)
- <eref> elements (Section 2.15)
- <iref> elements (Section 2.20)
- <cref> elements (Section 2.12)
- <spanx> elements (Section 2.36)
- <vspace> elements (Section 2.43)

2.38.1. ‘anchor’ attribute

Document-wide unique identifier for this paragraph.

The value needs to be a valid XML "Name" (Section 2.3 of [XML]).

2.38.2. ‘hangText’ attribute

Holds the label ("hanging text") for items in lists using the "hanging" style (see Section 2.22.3).

2.39. <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 a child element of: <section> (Section 2.34).

Content model:
In this order:
1. One optional <preamble> element (Section 2.29)
2. One or more <ttcol> elements (Section 2.41)
3. Optional <c> elements (Section 2.8)
4. One optional <postamble> element (Section 2.28)

2.39.1. ‘align’ attribute

Determines the horizontal alignment of the table.

Allowed values:
- "left"
- "center" (default)
- "right"

2.39.2. ‘anchor’ attribute

Document-wide unique identifier for this table.

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

The value needs to be a valid XML "Name" (Section 2.3 of [XML]).

2.39.3. ‘style’ attribute

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

Allowed values:
2.39.4. ‘suppress-title’ attribute

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

Allowed values:

- "true"
- "false" (default)

2.39.5. ‘title’ attribute

The title for the table; this usually appears on a line below the table body.

2.40. <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 a child element of: <front> (Section 2.19).

Content model: only text content.

2.40.1. ‘abbrev’ attribute

Specifies an abbreviated variant of the document title.

2.41. <ttcol>

Contains a column heading in a table.

This element appears as a child element of: <texttable>
(Section 2.39).
Content model: only text content.
2.41.1. ‘align’ attribute
Determines the horizontal alignment within the table column.
Allowed values:
  o "left" (default)
  o "center"
  o "right"
2.41.2. ‘width’ attribute
The desired column width (as integer 0..100 followed by "%").
2.42. <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 a child element of: <address> (Section 2.2).
Content model: only text content.
2.43. <vspace>
This element can be used to force the inclusion of a single line break or multiple blank lines.
Note that this is a purely presentational element and thus its use ought to be avoided.
This element appears as a child element of: <t> (Section 2.38).
Content model: this element does not have any contents.
2.43.1. ‘blankLines’ attribute
Number of blank lines to be inserted, where "0" indicates a single line break (defaults to "0").
For paged output formats, no additional blank lines should be
2.44. <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 2.33.9).

This element appears as a child element of: <front> (Section 2.19).

Content model: only text content.

2.45. <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.

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.

[[oi-xref: This needs to be expanded with examples and with a discussion how the autogenerated text differs when <xref> is not empty]]

This element appears as a child element of: <annotation> (Section 2.3), <c> (Section 2.8), <postamble> (Section 2.28), <preamble> (Section 2.29), and <t> (Section 2.38).

Content model: only text content.

2.45.1. 'format' attribute

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

Inserts a counter, such as the number of a section, figure, or table.

For targets that are not inherently numbered, such as references or comments, it uses the anchor name instead.

"default"

Inserts a text fragment that describes the referenced part completely, such as "Section 2", "Table 4", or "[XML]".

"none"

There will be no auto-generated text.

"title"

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

Allowed values:
- "counter"
- "title"
- "none"
- "default" (default)

2.45.2. ‘pageno’ attribute

Unused.

It’s unclear what the purpose of this attribute is; processors seem to ignore it and it never was documented.

Allowed values:
- "true"
- "false" (default)
2.45.3. ‘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. Escaping for Use in XML

Text in XML cannot use the literal characters "<" and ";", as they
have special meaning to the XML processor (starting entities,
elements, etc.). Usually, these characters will need to be
substituted by "&lt;" and "&amp;" (see Section 4.6 of [XML]).

">" does not require escaping, unless it appears in the sequence
"]]>" (which indicates the end of a CDATA section, see below).

Escaping the individual characters can be a lot of work (when done
manually), and also messes up alignment in artwork. Another approach
to escaping is to use CDATA sections ([XML], Section 2.7). Within
these, no further escaping is needed, except when the "end-of-CDATA"
marker needs to be used (in that case, the CDATA section needs to be
closed, and a new one needs to be started).

4. 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" -- 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
<?xml version="1.0"?>
<!DOCTYPE rfc [
  <!-- declare nbsp and friends -->
  <!ENTITY nbsp    "&#xa0;">  
  <!ENTITY nbhy    "&#x2011;">     
  <!ENTITY wj      "&#x2060;">  
]> 
```

5. Including Files

This version of the vocabulary does not support an inclusion mechanism on its own -- thus, a document always needs to be self-contained.

That being said, some processors do support file inclusion using processing instructions (Section 2.6 of [XML] and Section 4.1.2 of [TCLReadme]).

Furthermore, XML itself allows inclusion of external content using the "internal subset" (Section 2.8 of [XML]). Unfortunately, this requires declaring the external data in the DTD upfront.

For instance:

```xml
<?xml version="1.0"?>
<!DOCTYPE rfc [
  <!-- allow later RFC2616 reference using "&rfc2616;" -->
  <!-- the data will be fetched from xml.resource.org -->
]> 
```

...declares the entity "rfc2616", which then can be used in the "references" section:

```xml
<references>
  &rfc2616;
</references>
```

Note that this mechanism only works for well-formed XML fragments;
thus any plain text that would need to be escaped in XML can’t be
included as-is.

6. 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
characters ([USASCII] and Section 3 of [RFC2223]). It is possible
that this rule will be relaxed in future revisions of the RFC format
(for instance, to allow non-ASCII characters in examples and contact
information). In that case, it is expected that the vocabulary will
be extended accordingly.

7. 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.

Furthermore, the nature of XML, plus vocabulary features such as
typed artwork, make it attractive to extract content from documents
for further processing, such for the purpose of checking syntax, or
computing/verifying examples. In the latter case, care needs to be
taken that only trusted content is processed.

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

8. IANA Considerations

8.1. Internet Media Type Registration

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

This document serves as the specification for the Internet media type
"application/rfc+xml". 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 Section 9.1 of [RFC7303].

Encoding considerations: Identical to those of "application/xml" as described in Section 9.1 of [RFC7303].

Security considerations: As defined in Section 7. In addition, as this media type uses the "+xml" convention, it inherits the security considerations described in Section 10 of [RFC7303].

Interoperability considerations: N/A

Published specification: This specification.

Applications that use this media type: Applications that 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 9.1 of [RFC7303].

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

Macintosh file type code(s): TEXT

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

Intended usage: COMMON

Restrictions on usage: N/A

Author: See Authors Section.
9. Acknowledgments

Thanks to everybody who reviewed this document and provided feedback and/or specification text, in particular Brian Carpenter, Elwyn Davies, Tony Hansen, Paul Hoffman, Henrik Levkowetz, Alice Russo, Tom Taylor, Jim Schaad, and Nico Williams.

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

10. References

10.1. Normative References


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

10.2. Informative References


[RFCPOLICY] RFC Editor, "RFC Editorial Guidelines and Procedures".


Appendix A. Front Page (‘Boilerplate’) Generation

Reschke Expires July 11, 2015 [Page 43]
A.1. The /rfc/@category Attribute

For RFCs, the category attribute (Section 2.33.1) 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 (Section 2.33.4). The values are not the result of a grand design, but remain simply for historic reasons. Of these values, only a few are currently in use; all others are supported by 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 (Section 2.33.9) 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", which 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.
Formatters 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 value should be used unless one of the more specific '*'trust200902' values is a better fit. It produces the text in Sections 6.a and 6.b of the TLP.

A.2.1.2. noModificationTrust200902

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

This document may not be modified, and derivative works of it may not be created, except to format it for publication as an RFC or to translate it into languages other than English.

*Note:* this clause is incompatible with RFCs that are published on the Standards Track.

A.2.1.3. noDerivativesTrust200902

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

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

*Note:* this clause is incompatible with RFCs.

A.2.1.4. pre5378Trust200902

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

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

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

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

A.2.2. Historic Values

A.2.2.1. Historic Values: ‘*trust200811’

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

A.2.2.2. Historic Values: ‘*3978’

The attribute values "full3978", "noModification3978" and "noDerivatives3978" are similar to their counterparts above, except that they use text specified in Section 5 of [RFC3978].

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 Section 5 of [RFC3667].

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 Section 10 of [RFC2026].

The special value "none" was also used back then, and denied the IETF any rights beyond publication as Internet-Draft.
A.3.  The /rfc/@submissionType Attribute

The RFC Editor publishes documents from different "document streams", of which the "IETF stream" is the most prominent one. Other streams are the "independent stream" (used for things such as administrative information or April 1st RFCs), the "IAB stream" (Internet Architecture Board) and the "IRTF stream" (Internet Research Task Force).

The values for the attribute are "IETF" (the default value), "independent", "IAB", and "IRTF".

Historically, this attribute did not affect the final appearance of RFCs, except for subtle differences in Copyright notices. Nowadays (as of [RFC5741]), the stream name appears in the first line of the front page, and it also affects the text in the "Status Of This Memo" section.

For current documents, setting submissionType attribute will have the following effect:

- For RFCs, the stream name appears in the upper left corner of the first page (in Internet Drafts, this is either "Network Working Group", or the value of the <workgroup> element).
- For RFCs, if affects the whole "Status Of This Memo" section (see Section 3.2.2 of [RFC5741]).
- For all RFCs and Internet Drafts, it determines whether the "Copyright Notice" mentions the Copyright on Code Components (see TLP, Section "Text To Be Included in IETF Documents").

A.4.  The /rfc/@consensus Attribute

For some of the publication streams (see Appendix A.3), the "Status Of This Memo" section depends on whether there was a consensus to publish (again, see Section 3.2.2 of [RFC5741]).

The consensus attribute ("yes"/"no", defaulting to "yes") can be used to supply this information. The effect for the various streams is:

- "independent" and "IAB": none.
- "IETF": mention that there was an IETF consensus.
- "IRTF": mention that there was a research group consensus (where the name of the research group is extracted from the <workgroup> element).
Appendix B. Changes from RFC 2629 ('v1')

B.1. RNC Schema Differences (to be removed in RFC before publication)


B.2. Removed Elements

The <appendix> element has been removed; to generate an appendix, place a <section> inside <back>.

B.3. Changed Defaults

Many attributes have lost their "default" value; this is to avoid having document semantics differ based on whether a DTD was specified and evaluated. Processors will handle absent values the way the default value was specified before.

B.4. Changed Elements

<artwork>: Has a set of new attributes: "name", "type", "src", "align", "alt", "width", and "height". (Section 2.5)

<author>: The <organization> element is now optional. The "role" attribute was added. (Section 2.6)

<county>: The requirement to use ISO 3166 codes was removed. (Section 2.11)

<date>: All attributes are now optional. (Section 2.13)

<figure>: Has a set of new attributes: "suppress-title", "src", "align", "alt", "width", and "height". (Section 2.17)

<iref>: Has a new "primary" attribute. (Section 2.20)

<list>: The "style" attribute isn't restricted to a set of enumerated values anymore. The "hangIndent" and "counter" attributes have been added. (Section 2.22)

<reference>: <annotation> allows adding prose to a reference. The "anchor" attribute has been made mandatory. (Section 2.30)

<references>: Can now appear multiple times, and carry a "title" attribute (so that normative and informative references can be split). (Section 2.31)
<rfc>: The "ipr" attribute has gained additional values. The attributes "consensus", "iprExtract", "submissionType", and "xml:lang" have been added. (Section 2.33)

<section>: The new "toc" attribute controls whether it will appear in the Table Of Contents. <iref> can now appear as direct child element. (Section 2.34)

<t>: The "anchor" attribute can now be used as well, however there are restrictions on how they can be referred to. (Section 2.38)

B.5. New Elements

The following elements have been added: <annotation> (Section 2.3), <c> (Section 2.8), <cref> (Section 2.12), <format> (Section 2.18), <spanx> (Section 2.36), <texttable> (Section 2.39), <ttcol> (Section 2.41).

Appendix C. Relax NG 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"
attribute iprExtract { xsd:IDREF }?,
[ a:defaultValue = "IETF" ]
attribute submissionType {
   "IETF" | "IAB" | "IRTF" | "independent"
}?,
attribute docName { text }?,
[ a:defaultValue = "en" ] attribute xml:lang { 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" }?,
   organization?,
   address?
}

organization =
element organization {
   attribute abbrev { text }?,
   text
}

address = element address { postal?, phone?, facsimile?, email?,
   uri? }

   postal = element postal { street+, (city | region | code | country)*
}
street = element street { text }
city = element city { text }
region = element region { text }
code = element code { text }
country = element country { text }
phone = element phone { text }
facsimile = element facsimile { text }
email = element email { 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 = "default" ]
    attribute toc { "include" | "exclude" | "default" }?,

t =
   element t {
      attribute anchor { xsd:ID }?,
      attribute hangText { text }?,
      text
         \list
         figure
         xref
         xref
         iref
         cref
         spanx
         vspace)*
   }
\list =
   element list {
      attribute style { text }?,
      attribute hangIndent { text }?,
      attribute counter { text }?,
      t+
   }
xref =
   element xref {
      attribute target { xsd:IDREF },
      [ a:defaultValue = "false" ] attribute pageno { "true" | "false"
   }?,
      [ a:defaultValue = "default" ]
      attribute format { "counter" | "title" | "none" | "default" }?,
      text
   }
xref =
   element xref {
      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
}

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 }?,
  iref*,
  preamble?,
  artwork,
  postamble?
}

preamble =
element preamble { (text | xref | eref | iref | cref | spanx)* }

artwork =
element artwork {
[ a:defaultValue = "preserve" ]
attribute xml:space { "default" | "preserve" }?,
[ 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 }?,
text*
)
postamble =
  element postamble { (text | xref | eref | iref | cref | spanx)* }

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" }?,
preamble?,
ttcol+,
c*,
postamble?
}

ttcol =
element ttcol {
  attribute width { text }?,
  [ a:defaultValue = "left" ]
  attribute align { "left" | "center" | "right" }?,
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+

Reschke                   Expires July 11, 2015                [Page 54]
reference =
  element reference {
    attribute anchor { xsd:ID },
    attribute target { text }?,
    front,
    seriesInfo*,
    format*,
    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)* }
To check a file "test.xml" using the RNC file "schema.rnc", run (from a command line prompt):

```
java -jar jing.jar -c schema.rnc test.xml
```

In good Unix tradition, no output means the file is valid.

Index

A
  abbrev attribute
    in organization element 21
    in title element 33
  abstract element 5, 49
    inside front 16
  address element 5, 49
    inside author 9
  align attribute
    in artwork element 7
    in figure element 14
    in texttable element 32
    in ttcol element 34
  alt attribute
    in artwork element 7
    in figure element 14
  anchor attribute
    in cref element 11
    in figure element 14
    in reference element 23
    in section element 28
    in t element 31
    in texttable element 32
  annotation element 5, 49
    inside reference 23
  application/rfc+xml Media Type 39
  area element 6, 49
    inside front 16
  artwork element 6, 49
    align attribute 7
    alt attribute 7
    height attribute 7
    inside figure 14
    name attribute 7
    src attribute 8
    type attribute 8
    width attribute 8
    xml:space attribute 8

Attributes
abbrev 21, 33
align 7, 14, 32, 34
alt 7, 14
anchor 11, 14, 23, 28, 31-32
blankLines 34
category 25
consensus 25
counter 18
day 12
docName 25
format 35
fullname 9
hangIndent 18
hangText 31
height 7, 15
initials 9
ipr 26
iprExtract 26
item 17
month 12
name 7, 29
number 27
obsoletes 27
octets 16
pageno 36
primary 17
role 9
seriesNo 27
source 12
src 8, 15
style 18, 30, 32
subitem 17
submissionType 27
suppress-title 15, 33
surname 10
target 13, 16, 23, 37
title 15, 20, 24, 28, 33
toc 28
type 8, 16
updates 27
value 29
width 8, 15, 34
xml:lang 28
xml:space 8, 30
year 13
author element 9, 49
fullname attribute 9
initials attribute 9
inside front 16
role attribute 9
surname attribute 10

B
back element 10, 49
inside rfc 24
blankLines attribute
in vspace element 34

C
c element 10, 49
inside texttable 32
category attribute
in rfc element 25
city element 11, 49
inside postal 21
code element 11, 49
inside postal 21
consensus attribute
in rfc element 25
counter attribute
in list element 18
country element 11, 49
inside postal 21
cref element 11, 49
anchor attribute 11
inside annotation 6
inside c 10
inside postamble 22
inside preamble 22
inside t 31
source attribute 12

D
date element 12, 49
day attribute 12
inside front 16
month attribute 12
year attribute 13
day attribute
in date element 12
docName attribute
in rfc element 25

E
Elements
abstract 5, 16
address 5, 9
annotation 5, 23
area 6, 16
artwork 6, 14
author 9, 16
back 10, 24
c 10, 32
city 11, 21
code 11, 21
country 11, 21
cref 6, 10-11, 22, 31
date 12, 16
email 5, 13
eref 6, 10, 13, 22, 31
facsimile 5, 13
figure 14, 28, 31
format 15, 23
front 16, 23-24
iref 6, 10, 14, 16, 22, 28, 31
keyword 16-17
list 18, 31
middle 20, 24
note 16, 20
organization 9, 20
phone 5, 21
postal 5, 21
postamble 14, 22, 32
preamble 14, 22, 32
reference 23-24
references 10, 23
region 21, 24
rfc 24
section 10, 20, 28
seriesInfo 23, 29
spanx 6, 10, 22, 29, 31
street 21, 30
t 5, 18, 20, 28, 30
texttable 28, 31
title 16, 33
ttcol 32-33
uri 5, 34
vspace 31, 34
workgroup 16, 35
xref 5, 10, 22, 31, 35
email element 13, 49
inside address 5
eref element 13, 49
inside annotation 6
inside c 10
inside postamble 22
inside preamble 22
inside t 31
target attribute 13

F
facsimile element 13, 49
inside address 5
figure element 14, 49
align attribute 14
alt attribute 14
anchor attribute 14
height attribute 15
inside section 28
inside t 31
src attribute 15
suppress-title attribute 15
title attribute 15
width attribute 15
format attribute
in xref element 35
format element 15, 49
inside reference 23
octets attribute 16
target attribute 16
type attribute 16
front element 16, 49
inside reference 23
inside rfc 24
fullname attribute
in author element 9

H
hangIndent attribute
in list element 18
hangText attribute
in t element 31
height attribute
in artwork element 7
in figure element 15

I
initials attribute
in author element 9
ipr attribute
‘*2026’ 46
‘*3667’ 46
'*3978' 46
'*trust200811' 46
'*trust200902' 44
'noDerivativesTrust200902' 45
'noModificationTrust200902' 45
'pre5378Trust200902' 45
'trust200902' 45
in rfc element 26
iprExtract attribute
in rfc element 26
iref element 16, 49
inside annotation 6
inside c 10
inside figure 14
inside postamble 22
inside preamble 22
inside section 28
inside t 31
item attribute 17
primary attribute 17
subitem attribute 17
item attribute
in iref element 17

K
keyword element 17, 49
inside front 16

L
list element 18, 49
counter attribute 18
hangIndent attribute 18
inside t 31
style attribute 18
list styles
empty 18
format ... 19
hanging 19
letters 19
numbers 19
symbols 19

M
Media Type
application/rfc+xml 39
middle element 20, 49
inside rfc 24
month attribute
in date element 12

N
  name attribute
    in artwork element 7
    in seriesInfo element 29
  note element 20, 49
    inside front 16
    title attribute 20
  number attribute
    in rfc element 27

O
  obsoletes attribute
    in rfc element 27
  octets attribute
    in format element 16
  organization element 20, 49
    abbrev attribute 21
    inside author 9

P
  pageno attribute
    in xref element 36
  phone element 21, 49
    inside address 5
  postal element 21, 49
    inside address 5
  postamble element 22, 49
    inside figure 14
    inside texttable 32
  preamble element 22, 49
    inside figure 14
    inside texttable 32
  primary attribute
    in iref element 17

R
  reference element 23, 49
    anchor attribute 23
    inside references 24
    target attribute 23
  references element 23, 49
    inside back 10
    title attribute 24
  region element 24, 49
    inside postal 21
  rfc element 24, 49
category attribute 25
consensus attribute 25
docName attribute 25
ipr attribute 26
iprExtract attribute 26
number attribute 27
obsoletes attribute 27
seriesNo attribute 27
submissionType attribute 27
updates attribute 27
xml:lang attribute 28
role attribute
  in author element 9

section element 28, 49
anchor attribute 28
inside back 10
inside middle 20
inside section 28
title attribute 28
toc attribute 28
seriesInfo element 29, 49
inside reference 23
name attribute 29
value attribute 29
seriesNo attribute
  in rfc element 27
source attribute
  in cref element 12
spanx element 29, 49
inside annotation 6
inside c 10
inside postamble 22
inside preamble 22
inside t 31
style attribute 30
xml:space attribute 30
src attribute
  in artwork element 8
  in figure element 15
street element 30, 49
inside postal 21
style attribute
  in list element 18
  in spanx element 30
  in texttable element 32
subitem attribute
in iref element 17
submissionType attribute
in rfc element 27
suppress-title attribute
in figure element 15
in texttable element 33
surname attribute
in author element 10

T
t element 30, 49
   anchor attribute 31
   hangText attribute 31
   inside abstract 5
   inside list 18
   inside note 20
   inside section 28
   target attribute
   in eref element 13
   in format element 16
   in reference element 23
   in xref element 37
   texttable element 31, 49
   align attribute 32
   anchor attribute 32
   inside section 28
   style attribute 32
   suppress-title attribute 33
   title attribute 33
   title attribute
   in figure element 15
   in note element 20
   in references element 24
   in section element 28
   in texttable element 33
   title element 33, 49
   abbrev attribute 33
   inside front 16
toc attribute
   in section element 28
ttcol element 33, 49
   align attribute 34
   inside texttable 32
   width attribute 34
   type attribute
   in artwork element 8
   in format element 16
updates attribute
   in rfc element  27
   uri element  34, 49
       inside address  5

value attribute
   in seriesInfo element  29
   vspace element  34, 49
       blankLines attribute  34
       inside t  31

width attribute
   in artwork element  8
   in figure element  15
   in ttcol element  34
   workgroup element  35, 49
       inside front  16

xml:lang attribute
   in rfc element  28
xml:space attribute
   in artwork element  8
   in spanx element  30
xref element  35, 49
   format attribute  35
   inside annotation  5
   inside c  10
   inside postamble  22
   inside preamble  22
   inside t  31
   pageno attribute  36
   target attribute  37
xref formats
   counter  36
   default  36
   none  36
   title  36

year attribute
   in date element  13
Author’s Address

Julian F. Reschke
greenbytes GmbH
Hafenweg 16
Muenster, NW  48155
Germany

EMail: julian.reschke@greenbytes.de
URI:   http://greenbytes.de/tech/webdav/