

Network Working Group
Internet-Draft
Intended status: Informational
Expires: December 19, 2015

J. Hildebrand, Ed.
Cisco Systems, Inc.
P. Hoffman
VPN Consortium
June 17, 2015

HyperText Markup Language Request For Comments Format draft-hildebrand-html-rfc-07

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 December 19, 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. 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	4
2.1.	Requirements for Accessibility	6
3.	HTML Version	6
4.	HTML Syntax	7
5.	Common Items	7
5.1.	IDs	7
5.2.	Pilcrows	8
5.3.	ASCII Equivalents	8
6.	Front Matter	9
6.1.	DOCTYPE	9
6.2.	Root Element	9
6.3.	Head Element	9
6.3.1.	Charset Declaration	9
6.3.2.	Document Title	9
6.3.3.	Document metadata	10
6.3.4.	Style	10
6.3.5.	Links	11
6.4.	Document Information	11
6.5.	Table of Contents	11
7.	Main Body	12
8.	Back Matter	12
8.1.	Index	12
8.1.1.	Index Contents	12
8.1.2.	Index Letters	13
8.1.3.	Index Items	13
8.1.4.	Index Sub-items	14
8.2.	Authors' Addresses	14
8.3.	Document Information	15
8.4.	XML Source	16
9.	Elements	16
9.1.	<abstract>	16
9.2.	<address>	17
9.3.	<annotation>	17
9.4.	<area>	17
9.5.	<artwork>	17
9.5.1.	Text Artwork	17
9.5.2.	SVG Artwork	18
9.5.3.	Other Artwork	18
9.6.	<aside>	19
9.7.	<author>	19
9.8.	<back>	19
9.9.	<bcp14>	20

9.10.	<blockquote>	20
9.11.	<boilerplate>	20
9.12.	
	21
9.13.	<city>	21
9.14.	<code>	21
9.15.	<country>	21
9.16.	<cref>	21
9.17.	<date>	22
9.18.	<dd>	22
9.19.	<displayreference>	22
9.20.	<dl>	22
9.21.	<dt>	22
9.22.		22
9.23.	<email>	22
9.24.	<eref>	23
9.25.	<figure>	23
9.26.	<front>	23
9.27.	<ieref>	23
9.28.	<keyword>	23
9.29.		24
9.30.	<link>	24
9.31.	<middle>	24
9.32.	<name>	24
9.33.	<note>	24
9.34.		24
9.34.1.	Percent styles	25
9.34.2.	Standard styles	25
9.35.	<organization>	25
9.36.	<phone>	25
9.37.	<postal>	25
9.38.	<postalLine>	26
9.39.	<refcontent>	26
9.40.	<reference>	27
9.41.	<referencegroup>	27
9.42.	<references>	27
9.43.	<region>	28
9.44.	<relref>	28
9.44.1.	displayFormat='of'	28
9.44.2.	displayFormat='comma'	28
9.44.3.	displayFormat='parens'	29
9.44.4.	displayFormat='bare'	29
9.45.	<rfc>	30
9.46.	<section>	30
9.47.	<seriesInfo>	30
9.48.	<sourcecode>	30
9.49.	<street>	31
9.50.		31
9.51.	<sub>	31

9.52.	<sup>	31
9.53.	<svg>	31
9.54.	<t>	31
9.55.	<table>	32
9.56.	<tbody>	32
9.57.	<td>	32
9.58.	<tfoot>	32
9.59.	<th>	32
9.60.	<thead>	32
9.61.	<title>	32
9.62.	<tr>	32
9.63.	<tt>	32
9.64.		33
9.65.	<uri>	33
9.66.	<workgroup>	33
9.67.	<xref>	33
10.	IANA Considerations	33
11.	Security Considerations	34
12.	Acknowledgments	34
13.	References	34
13.1.	Normative References	34
13.2.	Informative References	34
	Authors' Addresses	35

[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.

[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](#)].

- o 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.

- o 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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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.
- o 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.

- o 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.
- o 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.
- o 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.
- o 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

- o 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
- o Specific instances where goals for accessibility are important in the design choices of the format have been called out in the text.
- o 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:

- o The HTML output is encoded as UTF-8, as specified in [[RFC3629](#)].
- o The document is valid HTML.
- o Double quotes (U+0022 QUOTATION MARK: ") are used to quote attribute values unless the HTML specification forbids quoting a particular attribute.
- o Each logical line is terminated solely with a \n (U+000A: LINE FEED), otherwise known as "Unix-style" line endings.
- o 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) 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.
- o Comments in the source XML, if any, will not be copied into the HTML.
- o 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 "¶". The pilcrow will be 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:

```
<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.
    <a href="#p-1.2-2" class="pilcrow">&para;</a></p>
  <!-- NO pilcrow here -->
</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 included in the HTML as in a tag with the class "ascii". The is included inside the tag that is showing the text to the user. For example:

```
<span class="surname">
  HILDEBRAND
  <span class="ascii">Hildebrand</span>
</span>
```


[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".

[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:

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

For example:

```
<meta name="author" content="Joe Hildebrand,Heather Flanagan">
<meta name="keywords" content="html,css,rfc">
<meta name="description" content="This document defines...">
<meta name="generator" content="xmljade v0.2.4">
```

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:

```
<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>
      (<span class="organization">Cisco Systems, Inc.</span>)</div>
    <div class="author">
      <span class="initial">H.</span>
      <span class="surname">Flanagan</span>
      (<span class="organization">RFC Editor</span>)</div>
  </dd>
</dl>
```

6.5. Table of Contents

The table of contents will follow the boilerplate if the XML's <rfc> element's tocInclude attribute has the value "true". An HTML <h2> heading containing the text "Table of Contents" will be followed by a <nav> element that contains a element for each depth of the section hierarchy. Each section will be represented by a 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>, , and 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>
    <ul class="toc">
      <li class="toc">
        <a href="s-1.1">1.1</a>. <a href="n-sub-intro">Sub Intro</a>
      </li>
    </ul>
  </ul>
  ...

```

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:

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

```


8.1.2. Index Letters

The index index is followed by a tag that contains a tag for each first letter represented in the index. This 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 tag also includes a tag that will contain the index items.

For example:

```
<ul>
  <li class="indexChar">
    <a href="#rfc.index.C" id="rfc.index.C">C</a>
    <ul>
      <!-- items go here -->
    </ul>
  </li>
  ...
```

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 tag of class "indexItem" containing a 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 tag with class "indexComma".

For example:

```
<li class="indexItem">
  <span class="irefItem">Bullets</span>
  <span class="irefRefs">
    <a class="indexPrimary" href="#s-Bullets-1">2</a>
    <span class="indexComma">,</span>
    <a href="#s-Bullets-2">2</a>
  </span>
  <!-- subitems go here -->
</li>
...
```


8.1.4. Index Sub-items

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

For example:

```
<ul>
  <li class="indexSubItem">
    <span class="irefSubItem">Ordered</span>
    <span class="irefRefs">
      <a href="#s-Bullets-Ordered-1">2</a>
    </span>
  </li>
</ul>
...
```

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.

```
<section id="author-addresses">
  <h2>
    <a class="self-ref" href="#author-addresses">
      Authors' Addresses
    </a>
  </h2>
  <address class="vcard">
    <div class="namerole">
      <span class="fn">
        Joe Hildebrand <span class="ascii">Joe Hildebrand</span>
      </span>
      <span class="anticomma">(</span>
      <span class="role">editor</span>
      <span class="comma">)</span>
    </div>
    <div class="org">
      Cisco Systems, Inc. <span class="ascii">Cisco Systems,
      Inc.</span>
    </div>
  </address>
</section>
```



```
<div class="adr">
  <div class="street-address">1 Main Street</div>
  <div class="street-address">Suite 1</div>
  <div>
    <span class="city">
      Denver <span class="ascii">Denver</span>
    </span>
    <span class="comma">,</span>
    <span class="region">CO</span>
    <span class="code">
      80202 <span class="ascii">80202</span>
    </span>
  </div>
  <div class="country-name">US</div>
</div>
<div>
  <span>Phone:</span>
  <span class="tel">+1-720-555-1212</span>
  <span class="type">VOICE</span>
</div>
<div>
  <span>Fax:</span>
  <span class="tel">+1-303-555-1212</span>
  <span class="type">fax</span>
</div>
<div>
  <span>Email:</span>
  <a class="email" href="mailto:jhildebr@example.com">
    jhildebr@example.com
  </a>
  <span class="ascii">jhildebr@example.com</span>
</div>
<div>
  <span>URI:</span>
  <a class="url" href="http://www.example.com">
    http://www.example.com
  </a>
</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.

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:

```
<div class="docinfo">
  <span class="finalized">
    Finalized: <time
      datetime="2015-04-29T18:59:08Z">2015-04-29T18:59:08Z</time>
    </span>
  <span class="rendered">
    Rendered: <time
      datetime="2015-04-29T18:59:10Z">2015-04-29T18:59:10Z</time>
    </span>
</div>
```

[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 "&#x2d;&#x2d;") -->
<!--
<?xml version="1.0" encoding="utf-8"?>
<rfc>
  <!--&#x2d;&#x2d; A comment &#x2d;&#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="self-ref">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.

```
<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.

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 <;, >;, and &;, 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>
  </div>
  <figcaption>
    <a href="#f-1">Figure 1.</a>
    <a class="self-ref" href="#n-it-figures" id="n-it-figures">
      It figures
    </a>
  </figcaption>
</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.

```

<div class="artwork art-svg" id="p-2-17">
  <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>
  <a href="#p-2-17" class="pilcrow">&para;</a></pre>
</div>

```

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.


```
<div class="artwork art-logo" id="p-2-58">
  
  <a class="pilcrow" href="#p-2-58">P.</a>
</div>
```

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.
  <a href="#p-1.2-6" class="pilcrow">&para;</a>
</aside>
```

9.7. <author>

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 element with CSS class "fn" containing the value of the "fullname" attribute of the <author> XML element, and an HTML 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 , if it exists.

```
<address class="vcard">
  <div class="namerole">
    <span class="fn">Joe Hildebrand</span>
    (<span class="role">editor</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.

9.8. <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 `` element with the CSS class "[bcp14](#)".

You `MUST` 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.


```
<section id="status-of-this-memo">
  <h2 id="s-boilerplate-1">
    <a href="#status-of-this-memo" class="self-ref">
      Status of this Memo</a>
    </h2>
    <p id="p-boilerplate-1-1">This Internet-Draft is submitted in full
      conformance with the provisions of BCP 78 and BCP 79.
      <a href="#p-boilerplate-1-1" class="pilcrow">&para;</a>
    </p>
  ...
```

[9.12.](#) `
`

This element is directly rendered as its HTML counterpart.

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

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

```
<span class='locality'>Guilford</span>
```

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

This element is rendered as a `` 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 `` 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 cref-source.

```
<span class="cref" id="crefAnchor">Just a brief comment
about something that we need to remember later.
<span class="cref-source">--life</span></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 `<front>` element, it gets the CSS class "published".

```
<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.

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

[9.21.](#) `<dt>`

This element is directly rendered as its HTML counterpart.

[9.22.](#) ``

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.


```
<div>
  <span>Email:</span>
  <a href="mailto:joe@example.net" class="email">joe@example.net</a>
  <span class='ascii'>joe@example.net</span>
</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 "self-ref" around the figure number. It will also contain another `<a>` element with CSS class "self-ref" around the figure name, if a name was given.

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


[9.29.](#) ``

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 "rfceditor-remove".

```
<section id="s-note-1" class="note rfceditor-remove">
  <h2>
    <a href="#n-editorial-note" class="self-ref">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.](#) ``

The output created from an `` 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 `` 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 `` 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 `` with CSS class "tel" containing the phone number and an HTML `` 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:

- o Each `<street>` is rendered
- o A `<div>` that includes:
 - * The rendering of all `<city>` elements
 - * A comma and whitespace
 - * The rendering of all `<region>` elements
 - * Whitespace
 - * The rendering of all `<code>` elements
- o The rendering of all `<country>` elements

```
<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="comma">,</span>
    <span class="region">CO</span>
    <span class="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">`.

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

[9.39.](#) `<refcontent>`

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

```
<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> ...</dd>
</dl>
```

9.41. <referencegroup>

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

```
<span class="referencegroup">
  <dl class="reference">...</dl>
</span>
```

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.

```
<section id="n-references">
  <h2 id="s-3">
    <a href="#s-3" class="self-ref">3.</a>
    <a href="#n-references" class="self-ref">References</a>
  </h2>
  <section id="n-informative-references">
    <h3 id="s-3.1">
      <a href="#s-3.1" class="self-ref">3.1.</a>
      <a href="#n-informative-references" class="self-ref">
        Informative References</a></h3>
      <dl class="reference">...
    </dl>
    </section>
  </section>
```


9.43. <region>

This element is rendered as a `` 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 [<a class="relref" href="#RFC7878">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 an `<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],
Section
2.3, 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 an `<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 [RFC7878]
(Section
2.3) 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 `Section 2.3` 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="self-ref">1.</a>
    <a href="#intro" class="self-ref">Introduction</a>
  </h2>
  <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 `` 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 `<`, `>`, and `&`, 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 <stdio.h>;

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.](#) ``

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".

```
<h1 id="title">HyperText Markup Language Request For  
  Comments Format</h1>
```

[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.](#) ``

This element is directly rendered as its HTML counterpart, with the minor exception that if the spacing attribute has the value "compact", a CSS class of "ulcompact" 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

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.

[W3C.REC-html5-20141028]
Hickson, I., Berjon, R., Faulkner, S., Leithead, T., Navara, E., O'Connor, E., and S. Pfeiffer, "HTML5", World Wide Web Consortium Recommendation REC-html5-20141028, October 2014, <<http://www.w3.org/TR/2014/REC-html5-20141028>>.

[W3C.REC-CSS2-20110607]
Bos, B., Celik, T., Hickson, I., and H. Lie, "Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification", World Wide Web Consortium Recommendation REC-CSS2-20110607, June 2011, <<http://www.w3.org/TR/2011/REC-CSS2-20110607>>.

[I-D.flanagan-rfc-framework]
Flanagan, H., "RFC Format Framework", [draft-flanagan-rfc-framework-03](#) (work in progress), January 2015.

13.2. Informative References

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

[RFC5646] Phillips, A. and M. Davis, "Tags for Identifying Languages", [BCP 47](#), [RFC 5646](#), September 2009.

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

[RFC7322] Flanagan, H. and S. Ginoza, "RFC Style Guide", [RFC 7322](#), September 2014.

[I-D.hoffman-xml2rfc]
Hoffman, P., "The 'XML2RFC' version 3 Vocabulary", [draft-hoffman-xml2rfc-19](#) (work in progress), June 2015.

[I-D.hoffman-rfcv3-preptool]
Hoffman, P. and J. Hildebrand, "RFC v3 Prep Tool Description", [draft-hoffman-rfcv3-preptool-03](#) (work in progress), June 2015.

Authors' Addresses

Joe Hildebrand (editor)
Cisco Systems, Inc.

Email: jhildebr@cisco.com

Paul Hoffman
VPN Consortium

Email: paul.hoffman@vpnc.org

