Network Working Group
Internet-Draft
Intended status: Informational
Expires: October 20, 2018

R. Tse
N. Nicholas
J. Lau
P. Brasolin
Ribose

April 18, 2018

AsciiRFC: Authoring Internet-Drafts And RFCs Using AsciiDoc
draft-ribose-asciirfc-08

Abstract

This document describes an AsciiDoc syntax extension called AsciiRFC, designed for authoring IETF Internet-Drafts and RFCs.

AsciiDoc is a human readable document markup language which affords more granular control over markup than comparable schemes such as Markdown.

The AsciiRFC syntax is designed to allow the author to entirely focus on text, providing the full power of the resulting RFC XML through the AsciiDoc language, while abstracting away the need to manually edit XML, including references.

This document itself was written and generated into RFC XML v2 (RFC7749) and RFC XML v3 (RFC7991) directly through asciidoctor-rfc, an AsciiRFC generator.

Status of This Memo

Copyright Notice

Table of Contents

1. Introduction

This document describes a markup language called "AsciiRFC", developed specifically for the purpose of generating RFC XML document, based on Asciidoctor syntax. AsciiRFC can be used to generate compliant RFC XML v2 and v3 documents through the usage of an open source, MIT-licensed Ruby gem called "asciidoctor-rfc" written by the authors [asciidoctor-rfc].

1.1. Designed for authoring RFCs and Internet-Drafts

Internet-Drafts and RFCs intended for publication submission to the IETF can be written in a multitude of formats today, including:

- XML: RFC XML v2 [RFC7749] and v3 [RFC7991]

- nroff: through [NroffEdit]

- o Microsoft Word: through usage of [RFC5385]
- o Lyx: through [lyx2rfc]
- o Pandoc: [RFC7328], through [pandoc2rfc] or [draftr]
- o Kramdown: through [kramdown-rfc2629]
- o mmark: through [mmark]

Interestingly, the last three are Markdown [RFC7763] variants.

As specified in [RFC7990], the IETF intends for the canonical format of RFCs to transition from plain-text ASCII to RFC XML v3 [RFC7991]. While plain-text will continue to be accepted from authors by the IETF, at least in the short- to medium-term, XML will be preferred for submission, and any plain-text submissions will need to be converted to RFC XML v3.

While this need is already met for RFC XML v2 [RFC7749] by the tools specified above, the transition to RFC XML v3 [RFC7991] places added onus on authors to generate compliant XML.

1.2. Relationship between AsciiRFC, AsciiDoc and Asciidoctor

[AsciiDoc] is a lightweight markup language and an alternative to Markdown, with features that make it attractive as a markup language for RFC with XML output.

[Asciidoctor] is an open source, MIT-licensed Ruby implementation of [AsciiDoc] that provides an enhancement of the original AsciiDoc markup language.

The variant of AsciiDoc syntax accepted by [Asciidoctor] is hereafter called "Asciidoctor syntax".

AsciiRFC, as described in this document, is an AsciiDoc variant developed on Asciidoctor syntax, created for the purpose of generating RFC XML documents.

1.3. Comparison with RFC XML tools based on Markdown variants

Section 1.2 of [RFC7764] famously states that "there is no such thing as 'invalid' Markdown, there is no standard demanding adherence to the Markdown syntax, and there is no governing body that guides or impedes its development." While there are contexts where that flexibility is useful, the authoring of RFCs does have a standard and a governing body, and there is such a thing as invalid RFC XML. A

more rigorous and extensible counterpart to Markdown, which still preserves its basic approach to formatting, can generate RFC XML that encompasses a fuller subset of the specification, and preempts malformed RFC XML output. The proposed markup language and associated Ruby gem has several advantages that we believe make it worth considering as an approach to generating RFC XML.

- AsciiDoc was designed from the beginning as a publishing language: it was initially intended as a plain-text alternative to the DocBook XML schema. For that reason, Asciidoctor natively supports the full range of formatting required by RFC XML (including notes, tables, bibliographies, source-code blocks, and definition lists), without resorting to embedded HTML or Markdown "flavours".

- AsciiRFC covers the full range of elements in RFC XML v2 and RFC XML v3.

- AsciiDoc in its Ruby-based Asciidoctor implementation is extensible, with a well-defined API. (Extensions have been harnessed to deal with bibliographic preprocessing for AsciiRFC.)

- AsciiRFC allows granular control of rendering, including user-specified attributes of text blocks.

- The Asciidoctor implementation allows document inclusion, for managing large-scale documentation projects.

- AsciiRFC allows granular control of permutations of block nesting, such as source code within lists or definition lists within unordered lists.

- As a more formal counterpart to Markdown, AsciiDoc is well-suited to generating XML that needs to conform to a specified schema. The asciidoctor-rfc Ruby gem developed around AsciiDoc validates its RFC XML output against the RelaxNG schema defined for RFC XML, and reports any issues to the end user.

- The asciidoctor-rfc Ruby gem incorporates validation not only of the XML structure of the standard, but also of the IETF workgroups it mentions against the latest listing online, and the bibliographic entries for RFC and other standards registered on <https://xml2rfc.tools.ietf.org>. The gem also dynamically fetches bibliographic entries from the xml2rfc site, and uses them to populate the RFC XML document.

As with Markdown, there is a wide range of tools that can render AsciiDoc; so AsciiRFC drafts of RFC documents can be previewed and

accessed without depending on the RFC tools ecosystem. Our realisation of RFC XML in AsciiRFC has aimed to ensure that, as much as possible, the markup language can be can be processed by generic Asciidoctor tools.

The only exception to this as an add-on is the optional bibliography module, which allows bibliographies to be assembled on the fly based on citations in a document: see Section 17.2.

2. Conventions Used in This Document

2.1. Terms and Definitions

The following terms and definitions apply to this document:

AsciiDoc
: The AsciiDoc markup language generically, as described in [AsciiDoc].

Asciidoctor syntax
: The enhanced AsciiDoc syntax implemented by the [Asciidoctor] Ruby gem.

AsciiRFC
: The AsciiDoc syntax developed for the purpose of generating RFC XML document based on Asciidoctor syntax, as described in this document.

2.2. Wrapping Lines in Documentation Examples

This document contains a number of examples as fragments of XML. In some cases, the examples contain URLs that are over 72 characters long and so need to be wrapped for presentation in this document even though they would not need to be wrapped in the actual source XML.

This document adopts a policy similar to that described in [I-D.wu-netmod-yang-xml-doc-conventions] to denote line wraps. When an XML example contains a URL, that URL is always included in double-quotes. If the URL would extend beyond 72 characters, the line is wrapped and the wrap is indicated with a backslash ("\"). The backslash appears without any additional space before the backslash and with no further characters after the backslash.

For example, the following XML...

```
<format type="TXT" target=
  "http://www.example.org/citations/cookpot.txt"/>
```

...may be presented as

```
<format type="TXT" target=
  "http://www.example.org/citations/\
  cookpot.txt"/>
```

3. Document Structure and Syntax

AsciiRFC consists of a subset of Asciidoctor syntax with the addition of bibliographic macros (Section 17.2). Asciidoctor syntax is presented in the Asciidoctor user manual [Asciidoctor-Manual].

AsciiRFC syntax consists of:

- o A document header, containing a title, a list of authors, and document attributes in lines prefixed with ":".

- o An optional document preamble, separated from the document header by a blank line, and not containing any section titles.

- o A number of sections, set off by a section title (a line prefixed with two or more "=".)

A section may contain:

- o Other sections, whose level of nesting is indicated by the number of "=" in their header.

- o Blocks of text. Blocks can have metadata (including a title, an anchor for cross-references, and attributes.)

Blocks can be:

- o Paragraphs, which are terminated by blank lines.

- o Lists. List items are by default paragraphs, but can span over multiple paragraphs.

- o Delimited blocks (with a line delimiter on either side of them); these include tables, notes, sidebars, source code, block quotes, examples, and unprocessed content (e.g. raw XML). Delimited blocks contain by default one or more paragraphs.

- o List items can contain other blocks, including both nested lists and delimited blocks.

- Some delimited blocks can contain other delimited blocks; for example, examples can contain source code as well as discussion in paragraphs.

- Blocks of text consist of inline text, which itself can contain markup.

Inline markup includes:

- Text formatting: Bold, italic, superscript, subscript, monospace.

- Markup macros: the "bcp14" and "comment" macros. (Asciidoctor syntax supports custom markup macros.)

- URLs, including display text to render.

- Inline anchors.

- Cross-references to anchors (IDs of blocks or spans of text), including display text to render.

- Images: SVG only, as specified in [RFC7996].

- Index terms.

3.1. Unsupported features compared with Asciidoctor syntax

Several features from Asciidoctor are not supported within AsciiRFC due to the lack of support within RFC XML, including:

- Audio and video files are not supported in AsciiRFC as today's RFC XML structure does not support audio or video.

- Equations are not supported as there is no current RFC XML equivalent. Asciidoctor provides native support for [AsciiMathML] and [TeX-LaTeX], via the [MathJax] tool.

- Footnotes are not supported in AsciiRFC as there is no equivalent semantic representation in RFC XML.

These carved out features can be easily supported by AsciiRFC once RFC XML allows these elements.

3.2. Mapping To RFC XML syntax

The Asciidoctor syntax document structure aligns with both the RFC XML v2 and the RFC XML v3 structure. In the following, RFC XML v3 equivalences are given to the basic Asciidoctor structure.

Header
"<rfc>" attributes, most "front" elements.

Preamble
"front/abstract" and "front/note".

Sections
"middle/section" elements.

Sections with bibliography style attribute
"back/references" elements.

Sections with appendix style attribute
"back/section" elements.

Paragraphs
"t" elements.

Lists
"ul", "ol", "dl" elements.

Delimited blocks
"artwork", "aside", "blockquote", "figure", "note", "sourcecode", "table".

Inline markup
"bcp14", "br", "cref", "em", "eref", "iref", "relref", "strong", "sub", "sup", "tt", "xref".

Full details of the mapping of AsciiRFC elements to RFC XML v2 and v3 elements, and of how to convert AsciiRFC documents to RFC XML, are given in the documentation of [asciidoctor-rfc].

3.3. Example Illustrations

This document utilizes example documents provided in Appendix A for demonstration of AsciiRFC syntax and usage. The source files and published versions (at the IETF Datatracker) of these example documents are available in Appendix A.

- "A Minimal Internet-Draft In AsciiRFC" Appendix A.1.1

- "The Holy Hand Grenade of Antioch" Appendix A.2.1

- "An API For Calendar-Based Fortune Heuristics Services" Appendix A.3.1

3.4. Simple Illustration

This section gives an overview of how to create an RFC XML document in AsciiRFC, with some pitfalls to be aware of.

Illustrations are in RFC XML v3 and RFC XML v2.

A sample AsciiRFC document is provided in Figure 1, together with its corresponding rendering in:

- RFC XML v3 (Figure 2)

- RFC XML v2 (Figure 3)

```
<CODE BEGINS>
= A Minimal Internet-Draft In AsciiRFC
:doctype: internet-draft
:name: draft-asciirfc-minimal-02
:abbrev: AsciiRFC Example
:status: informational
:ipr: trust200902
:submissionType: individual
:area: Internet
:intended-series: full-standard
:revdate: 2018-04-12T00:00:00Z
:fullname: Josiah Stinkney Carberry
:lastname: Carberry
:forename_initials: J. S.
:organization: Brown University
:phone: +1 401 863 1000
:street: Box K, 69 Brown Street
:city: Providence
:code: 02912
:country: United States of America
:uri: https://www.brown.edu
:email: josiah.carberry@ribose.com
:fullname_2: Truman Grayson
:lastname_2: Grayson
:forename_initials_2: T.
:organization_2: Brown University
:phone_2: +1 401 863 1000
:street_2: Box G, 69 Brown Street
:city_2: Providence
:code_2: 02912
:country_2: United States of America
:uri_2: https://www.brown.edu
:email_2: truman.grayson@ribose.com
```

```
[abstract]

This document provides a template on how to author (or migrate!)
a new Internet-Draft / RFC in the AsciiRFC format.

This template requires usage of the `asciidoctor-rfc` Ruby gem.

[#introduction]
== Introduction

AsciiRFC <<I-D.ribose-asciirfc>> is an extremely simple way to
author Internet-Drafts and RFCs without needing to manually
craft RFC XML conforming to <<RFC7991>>.

This is a template specifically made for authors to easily
start with creating an Internet-Draft conforming to <<RFC7991>>
and submittable to the IETF datatracker.

[#conventions]
== Terms and Definitions

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*",
"*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*",
"*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this
document are to be interpreted as described in BCP 14
<<RFC2119>> <<RFC8174>> when, and only when, they appear in
all capitals, as shown here.

This document also refers to the following terms and
definitions:

AsciiRFC::
an AsciiDoc-derived syntax used for authoring RFCs and
Internet-Drafts, as defined in <<I-D.ribose-asciirfc>>.

[#symbols]
== Symbols And Abbreviations

ADRFC::
abbreviated form of AsciiRFC


[#security]
== Security Considerations

Any security considerations should be placed here.

As described in <<main>> (here's how you refer a local anchor),
```

```
local tools have to be installed before the document template
can be built.

Running of these local tools *MAY* produce unintended side
effects that impact security.

[#iana]
== IANA Considerations

This document does not require any action by IANA.

But if it does, such as proposing changes to IANA registries,
please include them here.

[bibliography]
== Normative References

//bibliography::norm[]
++++

<reference anchor= 'RFC2119'
target= 'https://www.rfc-editor.org/info/rfc2119'>
<front>
<title>Key words for use in RFCs to Indicate Requirement
Levels</title>
<author initials= 'S.' surname= 'Bradner' fullname='S. Bradner'>
<organization />
</author>
<date year= '1997' month= 'March' />
<abstract>
<t>In many standards track documents several words are
used to signify the requirements in the specification.
These words are often capitalized. This document defines
these words as they should be interpreted in IETF
documents.  This document specifies an Internet Best
Current Practices for the Internet Community, and
requests discussion and suggestions for improvements.
</t>
</abstract>
</front>
<seriesInfo name= 'BCP' value= '14'/>
<seriesInfo name= 'RFC' value= '2119'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC2119'/>
</reference>

<reference anchor= 'RFC7991'
target= 'https://www.rfc-editor.org/info/rfc7991'>
<front>
```

```
<title>The &quot;xml2rfc&quot; Version 3 Vocabulary</title>
<author initials= 'P.' surname= 'Hoffman' fullname='P. Hoffman'>
<organization />
</author>
<date year= '2016' month= 'December' />
<abstract>
<t>This document defines the &quot;xml2rfc&quot;
version 3 vocabulary: an XML-based language used for
writing RFCs and Internet-Drafts.  It is heavily derived
from the version 2 vocabulary that is also under
discussion.  This document obsoletes the v2 grammar
described in RFC 7749.</t>
</abstract>
</front>
<seriesInfo name= 'RFC' value= '7991'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC7991'/>
</reference>

<reference anchor= 'RFC8174'
target= 'https://www.rfc-editor.org/info/rfc8174'>
<front>
<title>Ambiguity of Uppercase vs Lowercase in RFC 2119
Key Words</title>
<author initials= 'B.' surname= 'Leiba' fullname='B. Leiba'>
<organization />
</author>
<date year= '2017' month= 'May' />
<abstract>
<t>RFC 2119 specifies common key words that may be
used in protocol specifications.  This document aims to
reduce the ambiguity by clarifying that only UPPERCASE
usage of the key words have the defined special
meanings.</t>
</abstract>
</front>
<seriesInfo name= 'BCP' value= '14'/>
<seriesInfo name= 'RFC' value= '8174'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC8174'/>
</reference>
++++

[bibliography]
== Informative References

//bibliography::info[]
++++

<reference anchor= 'IETF.TLP'
```

```
  target= 'https://trustee.ietf.org/trust-legal-provisions.html'>
  <front>
  <title>IETF Trust Legal Provisions (TLP)</title>
  <author>
  <organization>IETF</organization>
  </author>
  <date month= 'April' day= '12' year='2018' />
  </front>
  </reference>

  <reference anchor= 'RNP' target= 'https://github.com/riboseinc/rnp/'>
  <front>
  <title>RNP: A C library approach to OpenPGP</title>
  <author>
  <organization>Ribose Inc.</organization>
  <address>
  <postal>
  <street>Suite 1111, 1 Pedder Street</street>
  <city>Central</city>
  <region>Hong Kong</city>
  <country>Hong Kong</country>
  </postal>
  <email>open.source@ribose.com</email>
  <uri>https://www.ribose.com</uri>
  </address>
  </author>
  <date day= '31' month= 'March' year='2018'/>
  </front>
  </reference>

  <reference anchor= 'I-D.ribose-asciirfc'>
  <front>
  <title>
  AsciiRFC: Authoring Internet-Drafts And RFCs Using AsciiDoc
  </title>
  <author initials= "R" surname= "Tse" fullname="Ronald Tse">
  <organization/>
  </author>
  <author initials= "J" surname= "Lau" fullname="Jeffrey Lau">
  <organization/>
  </author>
  <author initials= "N" surname= "Nicholas" fullname="Nick Nicholas">
  <organization/>
  </author>
  <author initials= "P" surname= "Brasolin" fullname="Paolo Brasolin">
  <organization/>
  </author>
  <date month= "March" day= "23" year="2018"/>
```

```
<abstract>
<t>This document describes an AsciiDoc syntax
extension called AsciiRFC, designed for authoring IETF
Internet-Drafts and RFCs.  AsciiDoc is a human readable document
markup language which affords more granular control over markup
than comparable schemes such as Markdown.  The AsciiRFC syntax
is designed to allow the author to entirely focus on text,
providing the full power of the resulting RFC XML through the
AsciiDoc language, while abstracting away the need to manually
edit XML, including references.  This document itself was
written and generated into RFC XML v2 (RFC7749) and RFC XML v3
(RFC7991) directly through asciidoctor-rfc, an AsciiRFC
generator.</t>
</abstract>
</front>
<seriesInfo name= "Internet-Draft" value= "draft-ribose-asciirfc-04"/>
<format type= "TXT" target=
"http://www.ietf.org/internet-drafts/draft-ribose-asciirfc-04.txt"/>
</reference>

<reference anchor= "RFC5378"
target="https://www.rfc-editor.org/info/rfc5378">
<front>
<title>Rights Contributors Provide to the IETF Trust</title>
<author initials= "S."
surname="Bradner" fullname="S. Bradner" role="editor">
<organization/>
</author>
<author initials= "J."
surname="Contreras" fullname="J. Contreras" role="editor">
<organization/>
</author>
<date year= "2008" month= "November"/>
<abstract><t>The IETF policies about rights in Contributions
to the IETF are designed to ensure that such Contributions
can be made available to the IETF and Internet communities
while permitting the authors to retain as many rights as
possible. This memo details the IETF policies on rights in
Contributions to the IETF. It also describes the
objectives that the policies are designed to meet. This
memo obsoletes RFCs 3978 and 4748 and, with BCP 79 and
RFC 5377, replaces Section 10 of RFC 2026. This document
specifies an Internet Best Current Practices for the
Internet Community, and requests discussion and
suggestions for improvements.</t></abstract>
</front>
<seriesInfo name= "BCP" value= "78"/>
<seriesInfo name= "RFC" value= "5378"/>
```

```
<seriesInfo name= "DOI" value= "10.17487/RFC5378"/>
</reference>

<reference anchor= 'RFC7253'
target= 'https://www.rfc-editor.org/info/rfc7253'>
<front>
<title>The OCB Authenticated-Encryption Algorithm</title>
<author initials= 'T.' surname= 'Krovetz' fullname='T. Krovetz'>
<organization />
</author>
<author initials= 'P.' surname= 'Rogaway' fullname='P. Rogaway'>
<organization />
</author>
<date year= '2014' month= 'May' />
<abstract><t>This document specifies OCB, a shared-key
blockcipher-based encryption scheme that provides
confidentiality and authenticity for plaintexts and
authenticity for associated data.  This document is a product
of the Crypto Forum Research Group (CFRG).</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '7253'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC7253'/>
</reference>
++++


[appendix]
[#appendix-a]
== Examples

=== Example 1

Here's an example of a properly wrapped code snippet in
accordance with rules specified in <<code-snippets>>.

[source,json]
----
<CODE BEGINS>
{
"code": {
"encoding": "ascii",
"type":     "rfc",
"authors":  [ "Josiah Carberry", "Truman Grayson" ]
}
}
<CODE ENDS>
----
```

```
[#acknowledgements]
== Acknowledgements

The authors would like to thank their families.
<CODE ENDS>
```

Figure 1: Sample Internet-Draft in AsciiRFC

The first block of text, from "= Template For Writing An Internet-Draft In AsciiRFC" through to ":email_2: thomas.kandell@brown.edu", is the document header. It contains a title in the first line, an author attribution ("Josiah Carberry; Thomas Kandell"), and then a set of document attributes, conveying information about the document as well as information about its authors. This information ends up in RFC XML either as attributes of the root "rfc" tag, elements of the "front" tag, or processing instructions.

The following blocks of text, up until the first section header ("== Introduction"), are the document preamble. They are treated by the document converter as containing the document abstract ("abstract"), followed by any notes if present.

Section headers delimit the sections of the main body of the document, starting with "== Introduction". The document converter treats the first section of the document as the start of the "middle" section in RFC XML. The first section header is followed by a paragraph, and other sections and paragraphs. The number of "=" signs can be one higher than that of the preceding section header, which indicates that they are subsections of that section; so "=== Operators" is a subsection of the preceding "== Symbols And Abbreviations".

The paragraphs contain some inline formatting (e.g. boldface: "*MUST*", monospace: "`type`"), and sections can also contain blocks other than normal paragraphs; the section "== Operators", for example, contains a definition list (whose terms are delimited by "::"), and the subsection "=== Example 1" contains a code snippet (delimited by "----", and tagged with the style attribute "[source,json]", indicating that this is a JSON sourcecode listing). The document can also include comments ("//" for inline, "////" for blocks), which are not rendered when the document is processed.

The introductory section in this example contains a citation of a reference, which in this version of AsciiRFC is treated identically to a cross-reference ("<<RFC7253>>") -- the crossreference being to the references section of the document. Sections and blocks of texts within the document can also be the target of crossreferences; for example, the section header "=== Operators" is preceded by the anchor

"[#operators]", and that anchor is already referenced in the section "== Security Considerations".

The third last and second last section are tagged with the style attribute "[bibliography]", which identifies them as references containers; the document converter accordingly inserts them into the "back" element under RFC XML. The contents of the references sections are in this instance raw XML, delimited as a passthrough block (with "++++"), which the converter does not alter.

The final section is tagged with the style attribute "[appendix]", and is treated as such.

The RFC XML v3 document generated from this AsciiRFC document is:

```
<CODE BEGINS>
<?xml version= "1.0" encoding= "US-ASCII"?>
<?xml-stylesheet type= "text/xsl" href= "rfc2629.xslt"?>
<!DOCTYPE rfc SYSTEM "rfc2629.dtd">
<?rfc strict= "yes"?>
<?rfc compact= "yes"?>
<?rfc subcompact= "no"?>
<?rfc toc= "yes"?>
<?rfc tocdepth= "4"?>
<?rfc symrefs= "yes"?>
<?rfc sortrefs= "yes"?>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
submissionType="IETF" prepTime="2018-04-18T03:35:29Z" version="3">
<front>
<title abbrev= "AsciiRFC Example">A Minimal Internet-Draft In
AsciiRFC</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="IETF" value="draft-asciirfc-minimal-02"/>
<seriesInfo name= "" status="full-standard"
value="draft-asciirfc-minimal-02"/>
<author fullname= "Josiah Stinkney Carberry" surname= "Carberry"
initials="J. S.">
<organization>Brown University</organization>
<address>
<postal>
<street>Box K, 69 Brown Street</street>
<city>Providence</city>
<code>02912</code>
<country>United States of America</country>
</postal>
<phone>+1 401 863 1000</phone>
<email>josiah.carberry@ribose.com</email>
<uri>https://www.brown.edu</uri>
```

```
</address>
</author>
<author fullname= "Truman Grayson" surname= "Grayson" initials="T.">
<organization>Brown University</organization>
<address>
<postal>
<street>Box G, 69 Brown Street</street>
<city>Providence</city>
<code>02912</code>
<country>United States of America</country>
</postal>
<phone>+1 401 863 1000</phone>
<email>truman.grayson@ribose.com</email>
<uri>https://www.brown.edu</uri>
</address>
</author>
<date day= "12" month= "April" year="2018"/>
<area>Internet</area>

<abstract><t>This document provides a template on how to author (or
migrate!)
a new Internet-Draft / RFC in the AsciiRFC format.</t>
<t>This template requires usage of the <tt>asciidoctor-rfc</tt> Ruby
gem.</t></abstract>
</front><middle>
<section anchor= "introduction" numbered=
"false"><name>Introduction</name><t>AsciiRFC <xref target=
"I-D.ribose-asciirfc"/> is an extremely simple way to
author Internet-Drafts and RFCs without needing to manually
craft RFC XML conforming to <xref target= "RFC7991"/>.</t>
<t>This is a template specifically made for authors to easily
start with creating an Internet-Draft conforming to <xref target=
"RFC7991"/>
and submittable to the IETF datatracker.</t></section>
<section anchor= "conventions" numbered= "false"><name>Terms and
Definitions</name><t>The key words "<bcp14>MUST</bcp14>", "<bcp14>MUST
NOT</bcp14>", "<bcp14>REQUIRED</bcp14>", "<bcp14>SHALL</bcp14>",
"<bcp14>SHALL NOT</bcp14>", "<bcp14>SHOULD</bcp14>", "<bcp14>SHOULD
NOT</bcp14>", "<bcp14>RECOMMENDED</bcp14>",
"<strong>NOT RECOMMENDED</strong>", "<bcp14>MAY</bcp14>", and
"<bcp14>OPTIONAL</bcp14>" in this
document are to be interpreted as described in BCP 14
<xref target= "RFC2119"/> <xref target="RFC8174"/> when, and only when,
they appear in
all capitals, as shown here.</t>
<t>This document also refers to the following terms and
definitions:</t>
<dl>
```

```
<dt>AsciiRFC</dt>
<dd>an AsciiDoc-derived syntax used for authoring RFCs and
Internet-Drafts, as defined in <xref target=
"I-D.ribose-asciirfc"/>.</dd>
</dl></section>
<section anchor= "symbols" numbered= "false">
<name>Symbols And Abbreviations</name>
<dl>
<dt>ADRFC</dt>
<dd>abbreviated form of AsciiRFC</dd>
</dl>
</section>
<section anchor= "main" numbered= "false"><name>Main
content</name><t>This is where you place the main content, and the
following
serves as a placeholder for your text.</t>
<t>Subsections are used here for demonstration purposes.</t>
<section anchor= "_getting_started" numbered= "false"><name>Getting
started</name><t>The AsciiRFC and RFC toolchains <bcp14>MUST</bcp14> be
available locally to
build this document template.</t>
<section anchor= "_asciirfc_toolchain" numbered= "false"><name>AsciiRFC
toolchain</name><t>You will need to have:</t>
<ul>
<li>Ruby: for running the AsciiRFC toolchain</li>
<li><tt>asciidoctor-rfc</tt> gem: for converting AsciiRFC into XML RFC
(v2 or v3)</li>
</ul></section>
<section anchor= "_xml_rfc_toolchain" numbered= "false"><name>XML RFC
toolchain</name><t>You will need to have:</t>
<ul>
<li>Python: for running <tt>xml2rfc</tt></li>
<li><tt>xml2rfc</tt>: for converting RFC XML (v2 or v3) into TXT</li>
<li><tt>idnits</tt>: for submission preflight</li>
</ul></section></section>
<section anchor= "_referencing_external_content" numbered= "false">
<name>Referencing external content</name>
<ul>
<li>This is a published RFC <xref target= "RFC7253"/></li>
<li>This is an Internet-Draft <xref target=
"I-D.ribose-asciirfc"/></li>
<li>This is an external reference <xref target= "RNP"/></li>
</ul>
</section>
<section anchor= "code-snippets" numbered= "false">
<name>Code snippets</name>
<t>Code snippets should be wrapped with <tt>&lt;CODE BEGINS&gt;</tt>
and
```

```
<tt>&lt;CODE ENDS&gt;</tt> blocks, as required by the IETF Trust Legal
Provisions (TLP) <xref target= "IETF.TLP"/> specified in <xref
target="RFC5378"/>.</t>
</section></section>
<section anchor= "security" numbered= "false"><name>Security
Considerations</name><t>Any security considerations should be placed
here.</t>
<t>As described in <xref target= "main"/> (here&#8217;s how you refer a
local anchor),
local tools have to be installed before the document template
can be built.</t>
<t>Running of these local tools <bcp14>MAY</bcp14> produce unintended
side
effects that impact security.</t></section>
<section anchor= "iana" numbered= "false"><name>IANA
Considerations</name><t>This document does not require any action by
IANA.</t>
<t>But if it does, such as proposing changes to IANA registries,
please include them here.</t></section>
</middle><back>
<references anchor= "_normative_references">
<name>Normative References</name>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2119.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7991.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8174.xml"
parse= "text"/>
</references>
<references anchor= "_informative_references">
<name>Informative References</name>
<reference anchor= "IETF.TLP" target=
"https://trustee.ietf.org/trust-legal-provisions.html">
<front>
<title>IETF Trust Legal Provisions (TLP)</title>
<author>
<organization>IETF</organization>
</author>
<date month= "April" day= "12" year="2018"/>
</front>
</reference>
<reference anchor= "RNP" target= "https://github.com/riboseinc/rnp/">
```

```
<front>
<title>RNP: A C library approach to OpenPGP</title>
<author>
<organization>Ribose Inc.</organization>
<address>
<postal>
<street>Suite 1111, 1 Pedder Street</street>
<city>Central</city>
<region>Hong Kong</region>
<country>Hong Kong</country>
</postal>
<email>open.source@ribose.com</email>
<uri>https://www.ribose.com</uri>
</address>
</author>
<date day= "31" month= "March" year="2018"/>
</front>
</reference>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml3/\
/reference.I-D.draft-ribose-asciirfc-04.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.5378.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7253.xml"
parse= "text"/>
</references>
<section anchor= "appendix-a" numbered= "false">
<name>Examples</name>
<section anchor= "_example_1" numbered= "false"><name>Example
1</name><t>Here&#8217;s an example of a properly wrapped code snippet in
accordance with rules specified in <xref target= "code-snippets"/>.</t>
<figure>
<sourcecode type= "json"><![CDATA[
<CODE BEGINS>
{
"code": {
"encoding": "ascii",
"type":     "rfc",
"authors":  [ "Josiah Carberry", "Truman Grayson" ]
}
}
<CODE ENDS>
]]></sourcecode>
```

```
</figure></section>
</section>
<section anchor= "acknowledgements" numbered= "false">
<name>Acknowledgements</name>
<t>The authors would like to thank their families.</t>
</section>
</back>
</rfc>
<CODE ENDS>
```

Figure 2: Sample Internet-Draft In AsciiRFC, Output In RFC XML v3 Format

Some default processing instructions have already been prefixed to the XML.

Our AsciiRFC converter can also generate RFC XML v2 from the same source AsciiRFC, as shown in Figure 3. Output in RFC XML v2 is not extensively described in this document.

```
<CODE BEGINS>
<?xml version= "1.0" encoding= "US-ASCII"?>
<?xml-stylesheet type= "text/xsl" href= "rfc2629.xslt"?>
<!DOCTYPE rfc SYSTEM "rfc2629.dtd" [
<!ENTITY RFC2119 SYSTEM
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2119.xml">
<!ENTITY RFC7991 SYSTEM
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7991.xml">
<!ENTITY RFC8174 SYSTEM
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8174.xml">
<!ENTITY I-D.ribose-asciirfc SYSTEM
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml3/\
/reference.I-D.draft-ribose-asciirfc-04.xml">
<!ENTITY RFC5378 SYSTEM
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.5378.xml">
<!ENTITY RFC7253 SYSTEM
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7253.xml">
]>
<?rfc strict= "yes"?>
<?rfc compact= "yes"?>
<?rfc subcompact= "no"?>
<?rfc toc= "yes"?>
<?rfc tocdepth= "4"?>
```

```
<?rfc symrefs= "yes"?>
<?rfc sortrefs= "yes"?>
<rfc ipr= "trust200902" category= "info" submissionType="IETF"
docName="draft-asciirfc-minimal-02">
<front>
<title abbrev= "AsciiRFC Example">A Minimal Internet-Draft In
AsciiRFC</title>
<author fullname= "Josiah Stinkney Carberry" surname= "Carberry"
initials="J. S.">
<organization>Brown University</organization>
<address>
<postal>
<street>Box K, 69 Brown Street</street>
<city>Providence</city>
<code>02912</code>
<country>United States of America</country>
</postal>
<phone>+1 401 863 1000</phone>
<email>josiah.carberry@ribose.com</email>
<uri>https://www.brown.edu</uri>
</address>
</author>
<author fullname= "Truman Grayson" surname= "Grayson" initials="T.">
<organization>Brown University</organization>
<address>
<postal>
<street>Box G, 69 Brown Street</street>
<city>Providence</city>
<code>02912</code>
<country>United States of America</country>
</postal>
<phone>+1 401 863 1000</phone>
<email>truman.grayson@ribose.com</email>
<uri>https://www.brown.edu</uri>
</address>
</author>
<date day= "12" month= "April" year="2018"/>
<area>Internet</area>

<abstract><t>This document provides a template on how to author (or
migrate!)
a new Internet-Draft / RFC in the AsciiRFC format.</t>
<t>This template requires usage of the <spanx style=
"verb">asciidoctor-rfc</spanx> Ruby gem.</t></abstract>
</front><middle>
<section anchor= "introduction" title= "Introduction"><t>AsciiRFC <xref
target= "I-D.ribose-asciirfc"/> is an extremely simple way to
author Internet-Drafts and RFCs without needing to manually
```

```
craft RFC XML conforming to <xref target= "RFC7991"/>.</t>
<t>This is a template specifically made for authors to easily
start with creating an Internet-Draft conforming to <xref target=
"RFC7991"/>
and submittable to the IETF datatracker.</t></section>
<section anchor= "conventions" title= "Terms and Definitions"><t>The key
words "<spanx style= "strong">MUST</spanx>", "<spanx style="strong">MUST
NOT</spanx>", "<spanx style="strong">REQUIRED</spanx>", "<spanx
style="strong">SHALL</spanx>",
"<spanx style= "strong">SHALL NOT</spanx>", "<spanx
style="strong">SHOULD</spanx>", "<spanx style="strong">SHOULD
NOT</spanx>", "<spanx style="strong">RECOMMENDED</spanx>",
"<spanx style= "strong">NOT RECOMMENDED</spanx>", "<spanx
style="strong">MAY</spanx>", and "<spanx
style="strong">OPTIONAL</spanx>" in this
document are to be interpreted as described in BCP 14
<xref target= "RFC2119"/> <xref target="RFC8174"/> when, and only when,
they appear in
all capitals, as shown here.</t>
<t>This document also refers to the following terms and
definitions:</t>
<t>
<list style= "hanging">
<t hangText= "AsciiRFC"><vspace blankLines="0"/>an AsciiDoc-derived
syntax used for authoring RFCs and
Internet-Drafts, as defined in <xref target=
"I-D.ribose-asciirfc"/>.</t>
</list>
</t></section>
<section anchor= "symbols" title= "Symbols And Abbreviations">
<t>
<list style= "hanging">
<t hangText= "ADRFC"><vspace blankLines="0"/>abbreviated form of
AsciiRFC</t>
</list>
</t>
</section>
<section anchor= "main" title= "Main content"><t>This is where you place
the main content, and the following
serves as a placeholder for your text.</t>
<t>Subsections are used here for demonstration purposes.</t>
<section anchor= "_getting_started" title= "Getting started"><t>The
AsciiRFC and RFC toolchains <spanx style= "strong">MUST</spanx> be
available locally to
build this document template.</t>
<section anchor= "_asciirfc_toolchain" title= "AsciiRFC
toolchain"><t>You will need to have:</t>
<t>
```

```
<list style= "symbols">
<t>Ruby: for running the AsciiRFC toolchain</t>
<t><spanx style= "verb">asciidoctor-rfc</spanx> gem: for converting
AsciiRFC into XML RFC
(v2 or v3)</t>
</list>
</t></section>
<section anchor= "_xml_rfc_toolchain" title= "XML RFC toolchain"><t>You
will need to have:</t>
<t>
<list style= "symbols">
<t>Python: for running <spanx style= "verb">xml2rfc</spanx></t>
<t><spanx style= "verb">xml2rfc</spanx>: for converting RFC XML (v2
or v3) into TXT</t>
<t><spanx style= "verb">idnits</spanx>: for submission preflight</t>
</list>
</t></section></section>
<section anchor= "_referencing_external_content" title= "Referencing
external content">
<t>
<list style= "symbols">
<t>This is a published RFC <xref target= "RFC7253"/></t>
<t>This is an Internet-Draft <xref target=
"I-D.ribose-asciirfc"/></t>
<t>This is an external reference <xref target= "RNP"/></t>
</list>
</t>
</section>
<section anchor= "code-snippets" title= "Code snippets">
<t>Code snippets should be wrapped with <spanx style= "verb">&lt;CODE
BEGINS&gt;</spanx> and
<spanx style= "verb">&lt;CODE ENDS&gt;</spanx> blocks, as required by
the IETF Trust Legal
Provisions (TLP) <xref target= "IETF.TLP"/> specified in <xref
target="RFC5378"/>.</t>
</section></section>
<section anchor= "security" title= "Security Considerations"><t>Any
security considerations should be placed here.</t>
<t>As described in <xref target= "main"/> (here&#8217;s how you refer a
local anchor),
local tools have to be installed before the document template
can be built.</t>
<t>Running of these local tools <spanx style= "strong">MAY</spanx>
produce unintended side
effects that impact security.</t></section>
<section anchor= "iana" title= "IANA Considerations"><t>This document
does not require any action by IANA.</t>
<t>But if it does, such as proposing changes to IANA registries,
```

```
please include them here.</t></section>
</middle><back>
<references title= "Normative References">
&RFC2119;
&RFC7991;
&RFC8174;
</references>
<references title= "Informative References">
<reference anchor= "IETF.TLP" target=
"https://trustee.ietf.org/trust-legal-provisions.html">
<front>
<title>IETF Trust Legal Provisions (TLP)</title>
<author>
<organization>IETF</organization>
</author>
<date month= "April" day= "12" year="2018"/>
</front>
</reference>
<reference anchor= "RNP" target= "https://github.com/riboseinc/rnp/">
<front>
<title>RNP: A C library approach to OpenPGP</title>
<author>
<organization>Ribose Inc.</organization>
<address>
<postal>
<street>Suite 1111, 1 Pedder Street</street>
<city>Central</city>
<region>Hong Kong</region>
<country>Hong Kong</country>
</postal>
<email>open.source@ribose.com</email>
<uri>https://www.ribose.com</uri>
</address>
</author>
<date day= "31" month= "March" year="2018"/>
</front>
</reference>
&I-D.ribose-asciirfc;
&RFC5378;
&RFC7253;
</references>
<section anchor= "appendix-a" title= "Examples">
<section anchor= "_example_1" title= "Example 1"><t>Here&#8217;s an
example of a properly wrapped code snippet in
accordance with rules specified in <xref target= "code-snippets"/>.</t>
<figure>
<artwork type= "json"><![CDATA[
<CODE BEGINS>
```

```
{
"code": {
"encoding": "ascii",
"type":     "rfc",
"authors":  [ "Josiah Carberry", "Truman Grayson" ]
}
}
<CODE ENDS>
]]></artwork>
</figure></section>
</section>
<section anchor= "acknowledgements" title= "Acknowledgements">
<t>The authors would like to thank their families.</t>
</section>
</back>
</rfc>
<CODE ENDS>
```

Figure 3: Sample Internet-Draft In AsciiRFC, Output In RFC XML v2 Format

4. Header And Document Attributes

The header gives the document title, followed by an optional author attribution, and a series of document attributes, with no empty lines.

4.1. Title And Basic Attributes

Document attributes are used to populate attributes of the root "rfc" element, "front" elements, and document-level processing instructions.

- ":doctype:" determines whether the document will be considered "rfc" or "internet-draft", and interprets other attributes accordingly.

- Certain attributes ("workgroup", "area", "keyword") are comma delimited, and result in repeated RFC XML elements.

Figure 4 demonstrates how to set the document header in AsciiRFC, with its rendering in RFC XML v3 shown in Figure 5.

```
<CODE BEGINS>
= The Holy Hand Grenade of Antioch
Arthur son of Uther Pendragon
:doctype: internet-draft
:abbrev: Hand Grenade of Antioch
:updates: 8140
:submission-type: independent
:name: draft-camelot-holy-grenade-01
:status: informational
:consensus: false
:area: General, Operations and Management
:keyword: rabbits, grenades, antioch, camelot
:ipr: trust200902
:toc-include: true
:sort-refs: true
:revdate: 2018-04-15T00:00:00Z
<CODE ENDS>
```

Figure 4: AsciiRFC Document Header

```
<CODE BEGINS>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
updates="8140" sortRefs="true" tocInclude="true"
submissionType="independent" prepTime="2018-04-18T03:35:33Z"
version="3">
<front>
<title abbrev= "Hand Grenade of Antioch">The Holy Hand Grenade of
Antioch</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="independent" value="draft-camelot-holy-grenade-01"/>
<author fullname= "Arthur son of Uther Pendragon" surname= "Pendragon"
initials="A.">
<organization>Camelot</organization>
<address>
<postal>
<street>Palace</street>
<street>Camel Lot 1</street>
<city>Camelot</city>
<keyword>grenades</keyword>
<keyword>camelot</keyword>

<abstract>

<!-- tag::preamble1[] -->

<CODE ENDS>
```

Figure 5: AsciiRFC Document Header Rendered As RFC XML v3

4.2. Detailed Author Information

The document header can spell out further information about authors, including contact details. The AsciiRFC header is shown in Figure 6 with its rendering in RFC XML v3 shown in Figure 7.

```
<CODE BEGINS>
= The Holy Hand Grenade of Antioch
Arthur son of Uther Pendragon
:doctype: internet-draft
:abbrev: Hand Grenade of Antioch
:updates: 8140
:submission-type: independent
:name: draft-camelot-holy-grenade-01
:status: informational
:consensus: false
:area: General, Operations and Management
:keyword: rabbits, grenades, antioch, camelot
:ipr: trust200902
:toc-include: true
:sort-refs: true
:revdate: 2018-04-15T00:00:00Z
:fullname: Arthur son of Uther Pendragon
:forename_initials: A.
:lastname: Pendragon
:email: arthur.pendragon@ribose.com
:organization: Camelot
:uri: http://camelot.gov.example
:street: Palace\ Camel Lot 1
:city: Camelot
:region: England
:country: United Kingdom
<CODE ENDS>
```

Figure 6: AsciiRFC Document Header With One Author

```
<CODE BEGINS>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
updates="8140" sortRefs="true" tocInclude="true"
submissionType="independent" prepTime="2018-04-18T03:35:33Z"
version="3">
<front>
<title abbrev= "Hand Grenade of Antioch">The Holy Hand Grenade of
Antioch</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="independent" value="draft-camelot-holy-grenade-01"/>
<author fullname= "Arthur son of Uther Pendragon" surname= "Pendragon"
initials="A.">
<organization>Camelot</organization>
<address>
<postal>
<street>Palace</street>
<street>Camel Lot 1</street>
<city>Camelot</city>
<region>England</region>
<country>United Kingdom</country>
</postal>
<email>arthur.pendragon@ribose.com</email>
<uri>http://camelot.gov.example</uri>
</address>
</author>
<date day= "15" month= "April" year="2018"/>
<area>General</area>
<area>Operations and Management</area>
<keyword>rabbits</keyword>
<keyword>grenades</keyword>
<keyword>antioch</keyword>
<keyword>camelot</keyword>

<abstract>

<!-- tag::preamble1[] -->

<CODE ENDS>
```

Figure 7: AsciiRFC Document Header With One Author (RFC XML v3)

Details of a second, third etc. author, including their organization and contact details, are provided by suffixing the relevant author attributes with "_2", "_3" etc., as shown in Figure 8 and its RFC XML v3 rendering Figure 9.

```
<CODE BEGINS>
= An API For Calendar-Based Fortune Heuristics Services
Gabriel Destiny; Charise Luck
:doctype: internet-draft
:abbrev: Calendar Fortune Heuristics API
:name: draft-divination-cfapi-00
:status: informational
:ipr: trust200902
:area: Internet
:submission-type: independent
:intended-series: informational
:revdate: 2018-03-23T00:00:00Z
:lastname: Destiny
:fullname: Gabriel Destiny
:forename_initials: G.
:organization: Divination Inc.
:email: gabriel.destiny@ribose.com
:street: 9288 N Divine Street
:city: Dunn
:code: 28334
:region: NC
:country: United States of America
:lastname_2: Luck
:fullname_2: Charise Luck
:forename_initials_2: C.
:organization_2: Divination Inc.
:email_2: charise.luck@ribose.com
:street_2: 9288 N Divine Street
:city_2: Dunn
:code_2: 28334
:region_2: NC
:country_2: United States of America
<CODE ENDS>
```

Figure 8

```
<CODE BEGINS>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
submissionType="independent" prepTime="2018-04-18T03:35:38Z"
version="3">
<front>
<title abbrev= "Calendar Fortune Heuristics API">An API For
Calendar-Based Fortune Heuristics Services</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="independent" value="draft-divination-cfapi-00"/>
<seriesInfo name= "" status="informational"
value="draft-divination-cfapi-00"/>
<author fullname= "Gabriel Destiny" surname= "Destiny" initials="G.">
<organization>Divination Inc.</organization>
<address>
<postal>
<street>9288 N Divine Street</street>
<city>Dunn</city>
<region>NC</region>
<code>28334</code>
<country>United States of America</country>
</postal>
<email>gabriel.destiny@ribose.com</email>
</address>
</author>
<author fullname= "Charise Luck" surname= "Luck" initials="C.">
<organization>Divination Inc.</organization>
<address>
<postal>
<street>9288 N Divine Street</street>
<city>Dunn</city>
<region>NC</region>
<code>28334</code>
<country>United States of America</country>
</postal>
<email>charise.luck@ribose.com</email>
</address>
</author>
<date day= "23" month= "March" year="2018"/>
<area>Internet</area>

<abstract>

<!-- tag::sample[] -->


<CODE ENDS>
```

Figure 9: AsciiRFC Document Header With Multiple Authors (RFC XML v3)

The initial author attribution in AsciiRFC, e.g. "Gabriel Destiny; Charlise Luck" in the example above, expects a strict format of First Name, zero or more Middle Names, Last name, and cannot process honorifics like "Dr." or suffixes like "Jr.".

Name attributes with any degree of complexity should be overriden by using the ":fullname:" and ":lastname:" attributes. The AsciiRFC ":forename_initials:" attribute replaces the built-in Asciidoctor syntax ":initials:" attribute (which includes the surname initial), and is not automatically populated from the name attribution.

4.3. XML Processing Information

A document header may also contain attribute headers which are treated as XML processing instructions. An AsciiRFC example is shown in Figure 10 with its rendering in Figure 11. (Note that several processing instructions are included by default in the output of the AsciiRFC processor.)

```
<CODE BEGINS>
= The Holy Hand Grenade of Antioch
Arthur son of Uther Pendragon
:doctype: internet-draft
:abbrev: Hand Grenade of Antioch
:updates: 8140
:submission-type: independent
:name: draft-camelot-holy-grenade-01
:status: informational
:consensus: false
:ipr: trust200902
:comments: yes
:notedraftinprogress: yes
<CODE ENDS>
```

Figure 10: AsciiRFC Document Header With XML Processing Information

```
<CODE BEGINS>
<?xml version= "1.0" encoding= "US-ASCII"?>
<?xml-stylesheet type= "text/xsl" href= "rfc2629.xslt"?>
<!DOCTYPE rfc SYSTEM "rfc2629.dtd">
<?rfc comments= "yes"?>
<?rfc notedraftinprogress= "yes"?>
<?rfc strict= "yes"?>
<?rfc compact= "yes"?>
<?rfc subcompact= "no"?>
<?rfc toc= "yes"?>
<?rfc tocdepth= "4"?>
<?rfc symrefs= "yes"?>
<?rfc sortrefs= "true"?>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
updates="8140" sortRefs="true" tocInclude="true"
submissionType="independent" prepTime="2018-04-18T03:35:33Z"
version="3">
<front>
<title abbrev= "Hand Grenade of Antioch">The Holy Hand Grenade of
Antioch</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="independent" value="draft-camelot-holy-grenade-01"/>
<author fullname= "Arthur son of Uther Pendragon" surname= "Pendragon"
initials="A.">
<organization>Camelot</organization>
<address>
<postal>
<street>Palace</street>
<street>Camel Lot 1</street>
<city>Camelot</city>
<keyword>grenades</keyword>
<CODE ENDS>
```

Figure 11: AsciiRFC Document Header With XML Processing Information (RFC XML v3)

In the foregoing, values for the processing instructions "strict", "compact", "toc" etc are included by default; but "comments" and "notedraftinprogress" are included as specified in the AsciiRFC document header. The default values provided for processing instructions can in fact be overriden through the AsciiRFC document header.

4.4. AsciiRFC-specific Document Attributes

A few document attributes are specific to the operation of the RFC XML document converter:

```
:no-rfc-bold-bcp14: false
   overrides the wrapping by default of boldface uppercase BCP14
   [RFC2119] words (e.g. "*MUST NOT*") with the "bcp14" element.

:smart-quotes: false
   overrides Asciidoctor's conversion of straight quotes and
   apostrophes to smart quotes and apostrophes.

:inline-definition-lists: true
   overrides the RFC XML v2 "idnits" requirement that a blank line be
   inserted between a definition list term and its definition.

<CODE BEGINS>
= The Holy Hand Grenade of Antioch
Arthur son of Uther Pendragon
:doctype: internet-draft
:status: informational
:name: draft-camelot-holy-grenade-00

== Section 1
The specification *MUST NOT* use the word _doesn't_.
<CODE ENDS>
```

Figure 12: AsciiRFC Document Header Without RFC-specific Attributes

```
<CODE BEGINS>
<rfc submissionType= "IETF" prepTime= "2017-11-25T10:23:39Z"
version="3">
<front>
<title>The Holy Hand Grenade of Antioch</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="IETF" value="draft-camelot-holy-grenade-00" />
<author fullname= "Arthur son of Uther Pendragon" surname=
"Pendragon"
initials="A.">
</author>
<date day= "25" month= "November" year="2017" />
</front>
<middle>
<section anchor= "_section_1" numbered= "false">
<name>Section 1</name>
<t>The specification  <bcp14>MUST NOT</bcp14>
use the word <em> doesn&#8217;t</em>.</t>
</section>
</middle>
</rfc>
<CODE ENDS>
```

Figure 13: AsciiRFC Document Header Without RFC-specific Attributes (RFC XML v3)

```
<CODE BEGINS>
= The Holy Hand Grenade of Antioch
Arthur son of Uther Pendragon
:doctype: internet-draft
:status: informational
:name: draft-camelot-holy-grenade-00
:no-rfc-bold-bcp14: false
:smart-quotes: false

== Section 1
The specification *MUST NOT* use the word _doesn't_.
<CODE ENDS>
```

Figure 14: AsciiRFC Document Header With Overridden RFC-specific Attributes

```
<CODE BEGINS>
<rfc submissionType= "IETF" prepTime= "2017-11-25T10:23:39Z"
version="3">
<front>
<title>The Holy Hand Grenade of Antioch</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="IETF" value="draft-camelot-holy-grenade-00" />
<author fullname= "Arthur son of Uther Pendragon" surname=
"Pendragon"
initials="A.">
</author>
<date day= "25" month= "November" year="2017" />
</front>
<middle>
<section anchor= "_section_1" numbered= "false">
<name>Section 1</name>
<t>The specification <strong>MUST NOT</strong>
use the word <em>doesn't</em>.</t>
</section>
</middle>
</rfc>
<CODE ENDS>
```

Figure 15: AsciiRFC Document Header With Overridden RFC-specific Attributes (RFC XML v3)

5. Preamble

The preamble in AsciiRFC is the text between the end of the document header (which terminates with a blank line) and the first section of text.

Any paragraphs of text in the preamble are treated as an abstract, and may optionally be tagged with the "abstract" style attribute.

Any notes in the preamble are treated as a "note" element.

An example of setting the preamble is given in Figure 16 with its rendering in Figure 17.

```
<CODE BEGINS>

[abstract]
The menagerie of beasts and artefacts depicted in RFC8140
may be usefully supplemented by other renowned figures of
Internet and more general lore. This document extends the
menagerie to the seminal fable of the
"Holy Hand Grenade of Antioch", as depicted in the
Monty Python film "Monty Python and the Holy Grail",
as well as "Spamalot", the musical inspired by the movie.

[NOTE,remove-in-rfc=false]
.Spamalot
The relevance of the musical "Spamalot" to Internet lore should be
obvious to the reader; but in case of doubt, see also
Section 1 ("What is Spam*?") of RFC2635.

<CODE ENDS>
```

Figure 16: AsciiRFC With Preamble

```
<CODE BEGINS>
<abstract>

<t>The menagerie of beasts and artefacts depicted in RFC8140
may be usefully supplemented by other renowned figures of
Internet and more general lore. This document extends the
menagerie to the seminal fable of the
"Holy Hand Grenade of Antioch", as depicted in the
Monty Python film "Monty Python and the Holy Grail",
as well as "Spamalot", the musical inspired by the
movie.</t></abstract><note removeInRFC= "false">
<name>Spamalot</name>
<t>The relevance of the musical "Spamalot" to Internet lore should be
obvious to the reader; but in case of doubt, see also
Section 1 ("What is Spam*?") of RFC2635.</t>
</note>

</abstract>
<CODE ENDS>
```

Figure 17: AsciiRFC With Preamble (RFC XML v3)

6. Sections and Paragraphs

Section headers are given with a sequence of "=", where the number of instances of "=" gives the header level. The document itself opens

with a single "=", and sections within it must be started with a minimum of "==".

Section numbering is toggled with the in-document attribute ":sectnums:" (on), ":sectnums!:" (off). The boolean "toc" attribute can also be set on sections, indicating whether the section can be included in the document's table of contents.

Figure 18 shows how sections and paragraphs are used in AsciiRFC, and its rendered form is shown in Figure 19.

```
<CODE BEGINS>

[toc=exclude]
:sectnums!:
== Terminology

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*",
"*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*",
"*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this document
are to be interpreted as described in BCP 14 <<RFC2119>> <<RFC8174>>
when, and only when, they appear in all capitals, as shown here.

:sectnums:
== Introduction

<<RFC8140>> refers to the intended move of RFC formatting to
XML2RFC v3 <<RFC7990>>, in the following terms:

<CODE ENDS>
```

Figure 18: AsciiRFC With Sections

```
<CODE BEGINS>

</front><middle>
<section anchor= "_terminology" toc= "exclude" numbered="false">
<name>Terminology</name>
<t>The key words "<bcp14>MUST</bcp14>", "<bcp14>MUST NOT</bcp14>",
"<bcp14>REQUIRED</bcp14>", "<bcp14>SHALL</bcp14>",
"<bcp14>SHALL NOT</bcp14>", "<bcp14>SHOULD</bcp14>", "<bcp14>SHOULD
NOT</bcp14>", "<bcp14>RECOMMENDED</bcp14>",
"<strong>NOT RECOMMENDED</strong>", "<bcp14>MAY</bcp14>", and
"<bcp14>OPTIONAL</bcp14>" in this document
are to be interpreted as described in BCP 14 <xref target= "RFC2119"/>
<xref target="RFC8174"/>
when, and only when, they appear in all capitals, as shown here.</t>
</section>
<section anchor= "_introduction" numbered=
"true"><name>Introduction</name><t><xref target= "RFC8140"/> refers to
the intended move of RFC formatting to
XML2RFC v3 <xref target= "RFC7990"/>, in the following terms:</t>

<CODE ENDS>
```

Figure 19: AsciiRFC With Sections (RFC XML v3)

Note that skipped sections, such as defining a section using "====" within a section defined using "==", is not allowed in AsciiRFC syntax. Doing so will trigger an error with the following message:

```
asciidoctor: WARNING: _filename_: line _X_:
  section title out of sequence: expected level 2, got level 3
```

7. Figures

AsciiRFC examples (corresponding to RFC XML Figures), source code Listings, and Literals (preformatted text) are all delimited blocks. Listings and Literals can occur nested within Examples.

An AsciiRFC example with a figure is given in Figure 20, and its rendering in Figure 21.

```
<CODE BEGINS>

[[killer-bunny]]
.A Photo Of The Killer Rabbit of Caerbannog Taken In Secret
====
[alt=The Killer Bunny, in ASCII]
....
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

```
   \\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
   \\\\\\\\\\\\\\\\\\\\\^^^^^^^^^^^^^^^^^^^^^^\\\\\\\\\\\\\\\\\
   \\\\\\\\\\\\\\\\\\\<<#MWSHARPMWMWMWTEETHWMWWM>>>\\\\\\\\\\\\
   \\\\\\\\\\\\\\\<<<#WMMWMWDEEPMDARKWCAVEMWWMMWM##>>>>\\\\\\\\
   \\\\\\\\\\\\\<<#WMWMWMWMWWM/^MWMWMWMWMWMW^WMWMWMMW#>>>\\\\\\
   \\\\\\\\\\\\<<#WMWMBEASTMW// \MWABBITWMW/ \MWMWMWMW##\\\\\\\
   \\\\\\\\\\##MWMWMMWMWMWMWM\\  \MWMWMWMW/  /MWMWMWMWM##\\\\\\
   \\\\\\\\##WMWMWMWMMWMWMWMWM\\  \MWMWMW/  /MWMWMWMMWMWMWM##\\
   \\\\\\\##MWMMRAVENOUSMWMWMWM\\  \====/  /MWMRABBITMWMWMWMW##
   \\\\\\##MWMWMWMWMMWMWMWMWMW[[            ]WMWMWMMWMWMWMWMWMW
   \\\\\##MWMWMWMWCARNIVOROUSW[[   3    3   ]MWMWTOOMDARKWMWMMW
   \\\\##MWMWDARKMWMWMWMWMWMWM//\     o    /MWMWMWMMWMWMWMMWMWM
   \\##MWMWMMKILLERABBITWMWMM//| \___vv___/ \WMPITCHWBLACKWMWMW
   \##MWMWMWMMWMWMWMWMWMMWMW// |   \-^^-/   |MWMWMWMMWMWMWMWMWM
   MWMWMWMMWMWVERYMDARKWMMW//  |            |MWMCAERBANNOGWMWMW
   MWMWMWMMWMWMWMWMWMWMWMM{{  /             /MWMWMMWMWMWMWMWMWM
   MULTRADARKWMWMHELPMWMWMW\\ \  |      |  |MWMCANMMWMWMWMMWMWW
   MWMWMWMWMMWMWMWMWMMWMWMWM\\ | |_     |  |_WMWMMYOUMWMMWWMWMW
   MWMMWMWMWMWMBLACKWMWMWMWWM\_|__-\-----\__-\MWMWMWMREADMWMWWM
   MWMWMWMMWMWMWMWMMWMWMWWMWMWMWMMWMWMWMWMWMWMWMWMWMWMWMMTHISWW
   MWVERYMMSCARYMWMWWMWMMWMWMWMWMWMWMWMWMWMWMWWMWMWMMWMWIWM'.',
   MWMWMMWMW======MWMMCANTWSEEMAMTHINGMMWMWMWMWMWMWMBETMMW` . `
   MWMWMWM// SKULL \MWMWMWMMWSCREAMMMWMWMWMMWMNOTMWMWMWW  ` . \
   MWMWMW|| |X||X| |MWMWCALLMMEWMMWMWMMWMWMWMWWM - ` ~ . , '
   MWMWMW||___ O __|MWMWMWMMWMWMWMWMMW'   ___________//   -_^_-
   MWMWMW \\||_|_||MWMW      '   . .     <_|_|_||_|__|     \O/
   MW   \\/\||v v||  -\\-------___     .   .,        \     |
       \\|  \_CHIN/  ==-(|CARROT/)\>     \\/||//         v\/||/
          )          /--------^-^            ,.          \|//
    #  \(/ .\\|x//                              " ' '
     . ,                \\||//        \||\\//    \\
   ....
   ====

   [[killer-source]]
   .C Code To Lure Killer Rabbit Back To Cave
   ====
   [source,c]
   ----
   <CODE BEGINS>
   /* Locate the Killer Rabbit */
   int type;
   unsigned char *killerRabbit =
   LocateCreature(&caerbannog, "killer rabbit");
   if( killerRabbit == 0 ){
   puts("The Killer Rabbit of Caerbannog is out of town.");
   return LOST_CREATURE;
   }
```

```
/* Load Cave */
unsigned char *cave = LoadPlace(&caerbannog,
"The Cave Of Caerbannog");
if( cave == 0 ){
puts("The Cave of Caerbannog must have moved.");
return LOST_PLACE;
}

/* Lure the Killer Rabbit back into the Cave */
unsigned char *carrot = allocateObjectInPlace(
carrot("fresh"), cave);
if( carrot == 0 ){
puts("No carrot, no rabbit.");
return LOST_LURE;
}

/* Finally, notify the Killer Rabbit to act */
return notifyCreature(killerRabbit, &carrot);
<CODE ENDS>
----
====


<CODE ENDS>
```

Figure 20: AsciiRFC With A Figure

```
<CODE BEGINS>

<figure anchor= "killer-bunny">
<name>A Photo Of The Killer Rabbit of Caerbannog Taken In
Secret</name>
<artwork type= "ascii-art" alt= "The Killer Bunny"><![CDATA[
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\^^^^^^^^^^^^^^^^^^^^^^\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\<<#MWSHARPMWMWMWTEETHWMWWM>>>\\\\\\\\\\\\
\\\\\\\\\\\\\\\<<<#WMMWMWDEEPMDARKWCAVEMWWMMWM##>>>>\\\\\\\\
\\\\\\\\\\\\\<<#WMWMWMWMWWM/^MWMWMWMWMWMW^WMWMWMMW#>>>\\\\\\
\\\\\\\\\\\\<<#WMWMBEASTMW// \MWABBITWMW/ \MWMWMWMW##\\\\\\\
\\\\\\\\\\##MWMWMMWMWMWMWM\\  \MWMWMWMW/  /MWMWMWMWM##\\\\\\
\\\\\\\\##WMWMWMWMMWMWMWMWM\\  \MWMWMW/  /MWMWMWMMWMWMWM##\\
\\\\\\\##MWMMRAVENOUSMWMWMWM\\  \====/  /MWMRABBITMWMWMWMW##
\\\\\\##MWMWMWMWMMWMWMWMWMW[[            ]WMWMWMMWMWMWMWMWMW
\\\\\##MWMWMWMWCARNIVOROUSW[[   3    3   ]MWMWTOOMDARKWMWMMW
\\\\##MWMWDARKMWMWMWMWMWMWM//\     o    /MWMWMWMMWMWMWMMWMWM
\\##MWMWMMKILLERABBITWMWMM//| \___vv___/ \WMPITCHWBLACKWMWMW
\##MWMWMWMMWMWMWMWMWMMWMW// |   \-^^-/   |MWMWMWMMWMWMWMWMWM
```

```
MWMWMWMMWMWVERYMDARKWMMW//  |            |MWMCAERBANNOGWMWMW
MWMWMWMMWMWMWMWMWMWMWMM{{  /             /MWMWMMWMWMWMWMWMWM
MULTRADARKWMWMHELPMWMWMW\\ \  |      |  |MWMCANMMWMWMWMMWMWW
MWMWMWMWMMWMWMWMWMMWMWMWM\\ | |_     |  |_WMWMMYOUMWMMWWMWMW
MWMMWMWMWMWMBLACKWMWMWMWWM\_|__-\-----\__-\MWMWMWMREADMWMWWM
MWMWMWMMWMWMWMWMMWMWMWWMWMWMWMWMWMWMWMWMWMWMWMWMWMWMMTHISWW
MWVERYMMSCARYMWMWWMWMMWMWMWMWMWMWMWMWMWMWMWMWMWMMWMWIWM'.',
MWMWMMWMW======MWMMCANTWSEEMAMTHINGMMWMWMWMWMWMWMBETMMW` . `
MWMWMWM// SKULL \MWMWMWMMWSCREAMMMWMWMWMMWMNOTMWMWMWW  ` . \
MWMWMW|| |X||X| |MWMWCALLMMEWMMWMWMMWMWMWMWWM - ` ~ . , '
MWMWMW||___ O __|MWMWMWMMWMWMWMWMMW'   ___________//   -_^_-
MWMWMW \\||_|_||MWMW      '   . .     <_|_|_||_|__|     \O/
MW   \\/\||v v||  -\\-------___     .   .,         \     |
    \\|  \_CHIN/  ==-(|CARROT/)\>     \\/||//         v\/||/
       )          /--------^-^            ,.            \|//
 #  \(/ .\\|x//                              " ' '
  . ,                \\||//        \||\\\//   \\
```

```
]]></artwork>
</figure>
<figure anchor= "killer-source">
<name>C Code To Lure Killer Rabbit Back To Cave</name>
<sourcecode type= "c"><![CDATA[
<CODE BEGINS>
/* Locate the Killer Rabbit */
int type;
unsigned char *killerRabbit =
LocateCreature(&caerbannog, "killer rabbit");
if( killerRabbit == 0 ){
puts("The Killer Rabbit of Caerbannog is out of town.");
return LOST_CREATURE;
}

/* Load Cave */
unsigned char *cave = LoadPlace(&caerbannog,
"The Cave Of Caerbannog");
if( cave == 0 ){
puts("The Cave of Caerbannog must have moved.");
return LOST_PLACE;
}

/* Lure the Killer Rabbit back into the Cave */
unsigned char *carrot = allocateObjectInPlace(
carrot("fresh"), cave);
if( carrot == 0 ){
puts("No carrot, no rabbit.");
return LOST_LURE;
}
```

```
/* Finally, notify the Killer Rabbit to act */
return notifyCreature(killerRabbit, &carrot);
<CODE ENDS>
]]></sourcecode>
</figure>

<CODE ENDS>
```

Figure 21: AsciiRFC With A Figure (RFC XML v3)

If an AsciiRFC Listing or Literal occurs outside of an Example (Figure 22), the RFC XML converter will supply the surrounding Figure element (Figure 23).

```
<CODE BEGINS>

[[hand-grenade-figure]]
.The Holy Hand Grenade of Antioch (don't pull the pin)
```

Figure 22: AsciiRFC With ASCII Art Without Figure Wrapping

```
                      ______
                     \\/  \/
                    __\\  /__
                   ||  //\   |
                   ||__\\/ __|
                      ||  |    ,---,
                      ||  |====`\  |
                      ||  |    '---'
                    ,--'*`--,
                  _||#|***|#|
              _,/.-'#|* *|#`-._
            ,,-'#####|   |#####`-.
          ,,'########|   |########`,
         //##########| o |##########\
        ||###########|   |###########|
       ||############| o |############|
       ||------------'   '------------|
       ||o  o  o  o  o   o  o  o  o  o|
        |-----------------------------|
        ||###########################|
         \\#########################/
          `..#####################,'
           ``..###############_,'
              ``--.._____..--'
                 `''-----''`
```

```
<CODE ENDS>



<CODE BEGINS>

<figure anchor= "hand-grenade-figure">
<name>The Holy Hand Grenade of Antioch (don't pull the pin)</name>
<artwork type= "ascii-art" alt= "Holy Hand Grenade of
Antioch"><![CDATA[
                        ______
                         \\/  \/
                        __\\  /__
                       ||  //\   |
                       ||__\\/ __|
                          ||  |    ,---,
                          ||  |====`\  |
                          ||  |    '---'
                        ,--'*`--,
                      _||#|***|#|
                   _,/.-'#|* *|#`-._
                 ,,-'#####|   |#####`-.
               ,,'########|   |########`,
              //##########| o |##########\
             ||###########|   |###########|
            ||############| o |############|
            ||------------'   '------------|
            ||o  o  o  o  o   o  o  o  o  o|
             |-----------------------------|
             ||###########################|
              \\#########################/
               `..#####################,'
                ``..###############_,'
                   ``--.._____..--'
                       `''-----''`
]]></artwork>
</figure>

<CODE ENDS>
```

Figure 23: AsciiRFC With ASCII Art Without Figure Wrapping (RFC XML v3)

8. Lists

8.1. Basic Lists

AsciiRFC supports ordered, unordered, and definition lists. Indentation of ordered and unordered lists is indicated by repeating the list item prefix ("*" and "." respectively); for definition lists, it is indicated by incrementing the number of definition term delimiters ("::").

List attributes can be used to specify the type of symbol used for ordered lists.

An example of an unordered list is shown in Figure 24 (with its rendered version in Figure 25). An example of an ordered list with ordered and unordered sublists is shown in Figure 26 (with its rendered version in Figure 27). An example of a definition list is shown in Figure 28 (with its rendered version in Figure 29).

```
<CODE BEGINS>

* Killed
** Sir Bors
** Sir Gawain
** Sir Ector
* Soiled Himself
** Sir Robin
* Panicked
** King Arthur
* Employed Ordnance
** The Lector
** Brother Maynard
* Scoffed
** Tim the Enchanter

<CODE ENDS>
```

Figure 24: AsciiRFC With Unordered lists

```
<CODE BEGINS>

<ul>
<li>
<t>Killed</t>
<ul>
<li>Sir Bors</li>
<li>Sir Gawain</li>
<li>Sir Ector</li>
</ul>
</li>
<li>
<t>Soiled Himself</t>
<ul>
<li>Sir Robin</li>
</ul>
</li>
<li>
<t>Panicked</t>
<ul>
<li>King Arthur</li>
</ul>
</li>
<li>
<t>Employed Ordnance</t>
<ul>
<li>The Lector</li>
<li>Brother Maynard</li>
</ul>
</li>
<li>
<t>Scoffed</t>
<ul>
<li>Tim the Enchanter</li>
</ul>
</li>
</ul>

<CODE ENDS>
```

Figure 25: AsciiRFC With Unordered Lists (RFC XML v3)

```
<CODE BEGINS>

. Preamble: St Attila Benediction
. Feast of the People on Sundry Foods
** Lambs
** Sloths
** Carp
** Anchovies
** Orangutangs
** Breakfast Cereals
** Fruit Bats
** _et hoc genus omne_
. Take out the Holy Pin
. The Count
[upperalpha]
.. Count is to Three: no more, no less
.. Not Four
.. Nor Two, except if the count then proceeds to Three
.. Five is Right Out
. Lob the Holy Hand Grenade of Antioch towards the Foe
. The Foe, being naughty in the *LORD's* sight, [bcp14]#shall# snuff it

<CODE ENDS>
```

Figure 26: AsciiRFC With Ordered lists

```
<CODE BEGINS>

<ol type= "1">
<li>Preamble: St Attila Benediction</li>
<li>
<t>Feast of the People on Sundry Foods</t>
<ul>
<li>Lambs</li>
<li>Sloths</li>
<li>Carp</li>
<li>Anchovies</li>
<li>Orangutangs</li>
<li>Breakfast Cereals</li>
<li>Fruit Bats</li>
<li>
<em>et hoc genus omne</em>
</li>
</ul>
</li>
<li>Take out the Holy Pin</li>
<li>
<t>The Count</t>
<ol type= "A">
<li>Count is to Three: no more, no less</li>
<li>Not Four</li>
<li>Nor Two, except if the count then proceeds to Three</li>
<li>Five is Right Out</li>
</ol>
</li>
<li>Lob the Holy Hand Grenade of Antioch towards the Foe</li>
<li>The Foe, being naughty in the <strong>LORD's</strong> sight,
<bcp14>SHALL</bcp14> snuff it</li>
</ol>

<CODE ENDS>
```

Figure 27: AsciiRFC With Ordered Lists (RFC XML v3)

```
<CODE BEGINS>

Holy Hand Grenade of Antioch::
Ordnance deployed by Brother Maynard under the incantation of a
lector, in order to dispense with the Foes of the Virtuous.
See <<hand-grenade-figure>>.

Holy Spear of Antioch::
A supposed relic of the crucifixion of Jesus Christ, this is one
of at least four claimed instances of the lance that pierced
Christ's side. Its historical significance lies in inspiring
crusaders to continue their siege of Antioch in 1098.

Sovereign's Orb of the United Kingdom::
Part of the Crown Jewels of the United Kingdom, the Sovereign's
Orb is a hollow gold sphere set with jewels and topped with a
cross.  It was made for Charles II in 1661. See <<sovereign-orb>>.

<CODE ENDS>
```

Figure 28: AsciiRFC With Definition lists

```
<CODE BEGINS>

<dl>
<dt>Holy Hand Grenade of Antioch</dt>
<dd>Ordnance deployed by Brother Maynard under the incantation of a
lector, in order to dispense with the Foes of the Virtuous.
See <xref target= "hand-grenade-figure"/>.</dd>
<dt>Holy Spear of Antioch</dt>
<dd>A supposed relic of the crucifixion of Jesus Christ, this is one
of at least four claimed instances of the lance that pierced
Christ's side. Its historical significance lies in inspiring
crusaders to continue their siege of Antioch in 1098.</dd>
<dt>Sovereign's Orb of the United Kingdom</dt>
<dd>Part of the Crown Jewels of the United Kingdom, the Sovereign's
Orb is a hollow gold sphere set with jewels and topped with a
cross.  It was made for Charles II in 1661. See <xref target=
"sovereign-orb"/>.</dd>
</dl>

<CODE ENDS>
```

Figure 29: AsciiRFC With Definition Lists (RFC XML v3)

8.2. List Continuation

A list item by default spans a single paragraph. A following paragraph or other block element can be appended to the current list item by prefixing it with "+" in a separate line.

See the "List Continuation" section in [Asciidoctor-Manual] for more information.

An example of list continuation with text is shown in Figure 30 with its rendered version in Figure 31.

```
<CODE BEGINS>

Trojan Rabbit::
In their siege of the French-occupied castle which may already
contain an instance of the Grail, Sir Bedevere the Wise proposes to
use a Trojan Rabbit to infiltrate the castle, with a raiding party
to take the French "not only by surprise, but totally unarmed."
+
The proposal, unsurprisingly, proved abortive. The more so as the
raiding party forgot to hide within the Trojan Rabbit, before the
French soldiers took the Trojan Rabbit inside the castle.

Killer Rabbit of Caerbannog::
Guarding the entrance to the Cave of Caerbannog; see <<caerbannog>>.

<CODE ENDS>
```

Figure 30: AsciiRFC List With Text Continuation

```
<CODE BEGINS>

<dl>
<dt>Trojan Rabbit</dt>
<dd>
<t>In their siege of the French-occupied castle which may already
contain an instance of the Grail, Sir Bedevere the Wise proposes to
use a Trojan Rabbit to infiltrate the castle, with a raiding party
to take the French "not only by surprise, but totally unarmed."</t>
<t>The proposal, unsurprisingly, proved abortive. The more so as the
raiding party forgot to hide within the Trojan Rabbit, before the
French soldiers took the Trojan Rabbit inside the castle.</t>
</dd>
<dt>Killer Rabbit of Caerbannog</dt>
<dd>Guarding the entrance to the Cave of Caerbannog; see <xref target=
"caerbannog"/>.</dd>
</dl>

<CODE ENDS>
```

Figure 31: AsciiRFC List With Text Continuation (RFC XML v3)

(Multiple paragraphs are not permitted within a list item in RFC XML v2. The RFC XML converter deals with this by converting paragraph breaks into line breaks within a list item.)

List continuations can also be embedded to populate a list item with a sequence of blocks as a unit (in an Asciidoctor syntax open block).

An example of list continuation with a delimited block is shown in Figure 32 with its rendered version in Figure 33.

```
<CODE BEGINS>

. Take out the Holy Pin
. The Count
+
----
integer count;
for count := 1 step 1 until 3 do
say(count)
comment Five is Right Out
----
. Lob the Holy Hand Grenade of Antioch towards the Foe
. Foe snuffs it

<CODE ENDS>
```

Figure 32: AsciiRFC List With Block Continuation

```
<CODE BEGINS>

<ol type= "1">
<li>Take out the Holy Pin</li>
<li>
<t>The Count</t>
<figure>
<sourcecode><![CDATA[
integer count;
for count := 1 step 1 until 3 do
say(count)
comment Five is Right Out
]]></sourcecode>
</figure>
</li>
<li>Lob the Holy Hand Grenade of Antioch towards the Foe</li>
<li>Foe snuffs it</li>
</ol>

<CODE ENDS>
```

Figure 33: AsciiRFC List With Block Continuation (RFC XML v3)

AsciiDoc, and thus AsciiRFC, considers paragraphs to be the basic level of blocks, and does not permit lists to be nested within them: any text after a list is considered to be a new paragraph.

Therefore, markup as shown in Figure 34 cannot be generated via AsciiRFC.

```
<CODE BEGINS>
<t>
This is the start of a paragraph.
<ul>
<li>List Entry 1</li>
<li>
<t>List Entry 2</t>
<t>Note 2.</t>
</li>
</ul>
And this is the continuation of the paragraph.
</t>
<CODE ENDS>
```

Figure 34: This RFC XML v3 Output Cannot Be Generated Using AsciiRFC

9. Blockquotes

Asciidoctor syntax supports blockquotes and quotations of verse; its block quotations permit arbitrary levels of quote nesting. RFC XML v3, and thus AsciiRFC, only supports one level of blockquotes.

Unlike RFC XML v2, RFC XML v3 does not support line breaks outside of tables, so verse quotations are converted to prose in the v3 converter.

An example of using AsciiRFC Blockquotes is given in Figure 35 with its rendered version in Figure 36.

```
<CODE BEGINS>

[quote,attribution="A. Farrel"]
____
Although the RFC Editor has recently dragged the IETF kicking and
screaming into the twentieth century [RFC7990] [RFC7996], there is a
yearning among all right-thinking Internet architects to "keep it
simple" and to return to the olden days when pigs could be given
thrust without anyone taking undue offence.
____

<CODE ENDS>
```

Figure 35: AsciiRFC Blockquote Usage

```
<CODE BEGINS>

<blockquote quotedFrom= "A. Farrel">
<t>Although the RFC Editor has recently dragged the IETF kicking and
screaming into the twentieth century [RFC7990] [RFC7996], there is a
yearning among all right-thinking Internet architects to "keep it
simple" and to return to the olden days when pigs could be given
thrust without anyone taking undue offence.</t>
</blockquote>

<CODE ENDS>
```

Figure 36: AsciiRFC Blockquote Usage (RFC XML v3)

10. Notes And Asides

Asciidoctor syntax supports a range of "admonitions", including notes, warnings, and tips. They are indicated by a paragraph prefix (e.g. "WARNING:"), or as a block with an admonition style attribute.

All admonitions are conflated in AsciiRFC, being converted to "note" elements in the document preamble, and "cref" elements in the main document.

This means that no admonitions will therefore appear in the textual output, unless forced to through the "comments" processing instruction. A sample admonition is shown in Figure 37, with its rendered output in Figure 38.

```
<CODE BEGINS>

[NOTE,display=true,source=Author]
====
Image courtesy of
https://camelot.gov.example/creatures-in-ascii/
====

<CODE ENDS>
```

Figure 37: An AsciiRFC Adminition Block

```
<CODE BEGINS>

<t><cref display= "true" source= "Author">Image courtesy of
<eref target=
"https://camelot.gov.example/creatures-in-ascii/"/></cref></t>

<CODE ENDS>
```

Figure 38: An AsciiRFC Adminition Block (RFC XML v3)

With RFC XML v2, note that no inline formatting is permitted for "cref" elements, and any such formatting is therefore stripped for v2 by the converter.

Because paragraphs in AsciiRFC cannot contain any other blocks, a comment at the end of a paragraph is treated as a new block. In the document converter, any such comments are moved inside the preceding RFC XML paragraph; if the comment is at the start of a section, as in the example above, it is wrapped inside a paragraph.

The RFC XML v3 converter also supports "asides" (Asciidoctor syntax Sidebars). A sample is shown in Figure 39, with its rendered output in Figure 40.

```
<CODE BEGINS>

****
While the exchange at the French-occupied castle is one of
the more memorable scenes of _Monty Python and the Holy Grail_,
the Trojan Rabbit has not reached the same level of cultural
resonance as its more murderous counterpart. Reasons for this
may include:

* Less overall screen-time dedicated to the Trojan Rabbit.

* The Trojan Rabbit as projectile has already been anticipated
by the Cow as projectile.
****

<CODE ENDS>
```

Figure 39: An AsciiRFC Sidebar Block

```
<CODE BEGINS>

<aside><t>While the exchange at the French-occupied castle is one of
the more memorable scenes of <em>Monty Python and the Holy Grail</em>,
the Trojan Rabbit has not reached the same level of cultural
resonance as its more murderous counterpart. Reasons for this
may include:</t>
<ul>
<li>Less overall screen-time dedicated to the Trojan Rabbit.</li>
<li>The Trojan Rabbit as projectile has already been anticipated
by the Cow as projectile.</li>
</ul></aside>

<CODE ENDS>
```

Figure 40: An AsciiRFC Sidebar Block Rendered As An Aside (RFC XML v3)

Comments given in AsciiDoc syntax (notated with initial "//") are not intended to be shown in the rendered output, and will not appear in the output as XML comments. XML comments can be generated by using the "[comment]#...#" inline formatting macro, or the "[.comment]" role attribute on blocks. A sample is shown in Figure 39 with its rendered output in Figure 40.

```
<CODE BEGINS>

The exchange of projectile animals was the beginning of a
long-running fruitful relationship between the British and the
French peoples,
[comment]#TODO: Will need to verify that claim.# which
arguably predates the traditional English enmity with the
French. [comment]#Strictly speaking, the Knights are Welsh.#

[.comment]
--
This document, as it turns out, has a profusion of XML comments.

As expected, they are ignored in any rendering of the document.
--


<CODE ENDS>
```

Figure 41: AsciiRFC delimited text intended as an XML Comment

```
<CODE BEGINS>

<t>The exchange of projectile animals was the beginning of a
long-running fruitful relationship between the British and the
French peoples,


<!-- TODO: Will need to verify that claim. -->

which
arguably predates the traditional English enmity with the
French.

<!-- Strictly speaking, the Knights are Welsh. -->

</t>


<!-- This document, as it turns out, has a profusion of XML comments.

As expected, they are ignored in any rendering of the document.
-->



<CODE ENDS>
```

Figure 42: AsciiRFC delimited text Rendered As An XML Comment (RFC XML v3)

11. Tables

AsciiRFC tables, like RFC XML v3, support distinct table heads, bodies and feet; cells spanning multiple rows and columns; and horizontal alignment. The larger range of table formatting options available in RFC XML v2 is also supported.

A sample of an AsciiRFC table is shown in Figure 43, with its rendered output in Figure 44.

Neither version of RFC XML is as expressive in its table structure as Asciidoctor syntax. RFC XML, for example, does not permit blocks within table cells.

```
<CODE BEGINS>

[grid=all,options="footer"]
|===
|French Castle | Cave of Caerbannog

2+|King Arthur
2+|Patsy
2+|Sir Bedevere the Wise
2+|Sir Galahad the Pure
2+|Sir Lancelot the Brave
2+|Sir Robin the Not-quite-so-brave-as-Sir-Lancelot
|French Guard with Outrageous Accent| Tim the Enchanter
|Other French Guards | Brother Maynard
| | The Lector
.3+^|not yet recruited
>|Sir Bors
>|Sir Gawain
>|Sir Ector

|Retinue of sundry knights
|Retinue of sundry more knights than at the French Castle
|===

<CODE ENDS>
```

Figure 43: An AsciiRFC Table

```
<CODE BEGINS>

<table>
<thead>
<tr>
<th align= "left">French Castle</th>
<th align= "left">Cave of Caerbannog</th>
</tr>
</thead>
<tbody>
<tr>
<td colspan= "2" align= "left">King Arthur</td>
</tr>
<tr>
<td colspan= "2" align= "left">Patsy</td>
</tr>
<tr>
<td colspan= "2" align= "left">Sir Bedevere the Wise</td>
</tr>
<tr>
```

```
<td colspan= "2" align= "left">Sir Galahad the Pure</td>
</tr>
<tr>
<td colspan= "2" align= "left">Sir Lancelot the Brave</td>
</tr>
<tr>
<td colspan= "2" align= "left">Sir Robin the
Not-quite-so-brave-as-Sir-Lancelot</td>
</tr>
<tr>
<td align= "left">French Guard with Outrageous Accent</td>
<td align= "left">Tim the Enchanter</td>
</tr>
<tr>
<td align= "left">Other French Guards</td>
<td align= "left">Brother Maynard</td>
</tr>
<tr>
<td align= "left"/>
<td align= "left">The Lector</td>
</tr>
<tr>
<td rowspan= "3" align= "center">not yet recruited</td>
<td align= "right">Sir Bors</td>
</tr>
<tr>
<td align= "right">Sir Gawain</td>
</tr>
<tr>
<td align= "right">Sir Ector</td>
</tr>
</tbody>
<tfoot>
<tr>
<td align= "left">Retinue of sundry knights</td>
<td align= "left">Retinue of sundry more knights than at the
French Castle</td>
</tr>
</tfoot>
</table>

<CODE ENDS>
```

Figure 44: An AsciiRFC Table (RFC XML v3)

12. Inline Formatting

12.1. Italics, Boldface, Monospace, Subscripts, Superscripts

AsciiRFC supports italics, boldface, monospace, subscripts and superscripts, just like RFC XML v3.

The inline formatting syntax given in Figure 45 produces the RFC XML v3 output given in Figure 46.

```
<CODE BEGINS>

The participants of that renowned exercise in cross-cultural
communication, to wit the exchange between the
_Knights of the Round Table_
and the taunting French soldiers serving under *Guy de Lombard* are,
properly speaking, outside the scope of this `menagerie`, being more
or less human. Notwithstanding, several^ish^ beasts both animate~d~
and wooden played a significant part in this encounter; most
notably:

* The Projectile Cow, see <<projectile-cow>>
* The Trojan Rabbit, see <<trojan-rabbit>>

<CODE ENDS>
```

Figure 45: Inline Formatting In AsciiRFC

```
<CODE BEGINS>

<t>The participants of that renowned exercise in cross-cultural
communication, to wit the exchange between the
<em>Knights of the Round Table</em>
and the taunting French soldiers serving under <strong>Guy de
Lombard</strong> are,
properly speaking, outside the scope of this <tt>menagerie</tt>, being
more
or less human. Notwithstanding, several<sup>ish</sup> beasts both
animate<sub>d</sub>
and wooden played a significant part in this encounter; most
notably:</t>
<ul>
<li>The Projectile Cow, see <xref target= "projectile-cow"/></li>
<li>The Trojan Rabbit, see <xref target= "trojan-rabbit"/></li>
</ul>

<CODE ENDS>
```

Figure 46: Inline Formatting In AsciiRFC (RFC XML v3)

12.2. Formatting BCP 14 Keywords

RFC XML v3 also supports tagging of BCP14 keywords [RFC2119] [RFC8174]; this is done in AsciiRFC either by tagging them with a custom formatting span (e.g. "MUST NOT"), or by converting any boldface all-caps words recognised as BCP14 words (unless the ":no-rfc-bold-bcp14: false" document attribute is set).

Any spans of BCP14 text delimited by inline formatting delimiters need to be contained within a single line of text; in Asciidoctor syntax, formatting spans are broken up across line breaks.

This usage is demonstrated in Figure 47 with the rendered output in Figure 48.

```
<CODE BEGINS>

The instructions in the _Book of Armaments_ on the proper deployment
of the Holy Hand Grenade of Antioch [bcp14]#may# be summarized as
follows, although this summary *SHALL NOT* be used as a substitute
for a reading from the Book of Armaments:

<CODE ENDS>
```

Figure 47: BCP14 Keywords In AsciiRFC

```
<CODE BEGINS>

<t>The instructions in the <em>Book of Armaments</em> on the proper
deployment
of the Holy Hand Grenade of Antioch <bcp14>MAY</bcp14> be summarized as
follows, although this summary <bcp14>SHALL NOT</bcp14> be used as a
substitute
for a reading from the Book of Armaments:</t>

<CODE ENDS>
```

Figure 48: BCP14 Keywords In AsciiRFC (RFC XML v3)

12.3. Escaping AsciiRFC Syntax

Formatting delimiters like "*" can be escaped with backslash ("*"); double formatting delimiters, like "**" and "__", need to be escaped with double backslash ("**").

Escaping delimiters is not always reliable, and for double delimiters it is preferable to use HTML entities ("**"), or attribute references (references to the value of attributes set in the document header) as shown in Figure 49.

```
<CODE BEGINS>
:dblast: **

`{dblast}`
<CODE ENDS>
```

Figure 49: Escaping AsciiRFC Syntax Using Attributes

In extreme circumstances (such as quoting AsciiDoc syntax), you may need to resort to altering the substitutions behaviour within a given block of of AsciiDoc; see the "Applying Substitutions" section of [Asciidoctor-Manual].

13. Links

Common URL formats are recognised automatically as hyperlinks in AsciiRFC, which inherits this excellent feature from AsciiDoc, and are rendered as such.

Any hyperlinked text is appended after the hyperlink in square brackets.

An example is given in Figure 50 with its rendered version in Figure 51.

```
<CODE BEGINS>

<<killer-bunny,The following depiction>> of the fearsome beast
has been sourced from
http://camelot.gov.example/avatars/rabbit[Rabbit-SCII],
<<killer-source,accompanied>>
by C code that was used in this accurate depiction of the
Killer Rabbit:

<CODE ENDS>
```

Figure 50: An AsciiRFC Link

```
<CODE BEGINS>

<t><xref target= "killer-bunny">The following depiction</xref> of the
fearsome beast
has been sourced from
<eref target=
"http://camelot.gov.example/avatars/rabbit">Rabbit-SCII</eref>,
<xref target= "killer-source">accompanied</xref>
by C code that was used in this accurate depiction of the
Killer Rabbit:</t>

<CODE ENDS>
```

Figure 51: An AsciiRFC Link (RFC XML v3)

To prevent hyperlinking of a URL, prefix it with a backslash, as shown in Figure 52 with its rendered version in Figure 53.

```
<CODE BEGINS>

The screaming move into the twenty-*first* century is accompanied by
a move back to the late twentieth century, with ASCII stylings more
wonted in haunts like \ftp://ftp.wwa.com/pub/Scarecrow (known to be
accessible in 1996.)

<CODE ENDS>
```

Figure 52: A Literal AsciiRFC Link

```
<CODE BEGINS>

<t>The screaming move into the twenty-<strong>first</strong> century is
accompanied by
a move back to the late twentieth century, with ASCII stylings more
wonted in haunts like ftp://ftp.wwa.com/pub/Scarecrow (known to be
accessible in 1996.)</t>

<CODE ENDS>
```

Figure 53: A Literal AsciiRFC Link (RFC XML v3)

14. Cross-References

14.1. Basic Referencing

Anchors for cross-references are notated as "[[...]]" or "[#...]", and can be inserted on their own line in front of most blocks.

Asciidoctor syntax supports anchors in a much wider range of contexts than is supported than RFC XML v3 (let alone v2); anchors that are not supported for that version of RFC XML are simply ignored by the converter.

Note that anchors in RFC XML are constrained to the format "[A-Za-z_:][[A-Za-z0-9_:.-]*" (i.e. "xsd:ID").

Cross-references to anchors are notated as "<<...>>"; cross-references with custom text as "<<reference,text>>".

An example of using cross-references in AsciiRFC is given in Figure 54 with its rendered output in Figure 55.

```
<CODE BEGINS>

The _Cave of Caerbannog_ has been well-established in the mythology
of Camelot (as recounted by Monty Python) as the lair of the
Legendary Black Beast of Arrrghhh, more commonly known today as the
*Killer Rabbit of Caerbannog* <<killer_rabbit_caerbannog>>.
It is the encounter between the Killer Rabbit of Caerbannog and the
Knights of the Round Table, armed with the Holy Hand Grenade of
Antioch (see the <<holy_hand_grenade,following section>>), that we
recount here through monospace font and multiple spaces.

[[killer_rabbit_caerbannog]]
=== The Killer Rabbit of Caerbannog

<CODE ENDS>
```

Figure 54: Setting And Referring To Cross-References In AsciiRFC

```
<CODE BEGINS>

<t>The <em>Cave of Caerbannog</em> has been well-established in the
mythology
of Camelot (as recounted by Monty Python) as the lair of the
Legendary Black Beast of Arrrghhh, more commonly known today as the
<strong>Killer Rabbit of Caerbannog</strong> <xref target=
"killer_rabbit_caerbannog"/>.
It is the encounter between the Killer Rabbit of Caerbannog and the
Knights of the Round Table, armed with the Holy Hand Grenade of
Antioch (see the <xref target= "holy_hand_grenade">following
section</xref>), that we
recount here through monospace font and multiple spaces.</t>
<section anchor= "killer_rabbit_caerbannog" numbered= "true"><name>The
Killer Rabbit of Caerbannog</name>
<CODE ENDS>
```

Figure 55: Setting And Referring To Cross-References In AsciiRFC (RFC XML v3)

14.2. Referencing With Attributes

While Asciidoctor syntax natively does not support attributes on cross-references, AsciiRFC works around that by embedding formatting information as templated text within cross-references:

o "format= x: text" populates the "xref@format" attribute

o a section number followed by one of the words "of", "parens", "bare", "text" is treated as a "relref" reference to an external document.

An example of referencing with attributes is given in Figure 56 with its output in Figure 57.

```
<CODE BEGINS>

The *Killer Rabbit of Caerbannog*, that most formidable foe of
the Knights and of all that is holy or carrot-like, has been
depicted diversely in lay and in song. We venture to say,
_contra_ the claim made in <<RFC8140,4.1 of: Ze Vompyre>>,
that the Killer Rabbit of Caerbannog truly is the most afeared
of all the creatures. Short of sanctified ordnance such as
<<holy_hand_grenade,format=title>>, there are few remedies
known against its awful lapine powers.

<CODE ENDS>
```

Figure 56: Cross-References With Attributes In AsciiRFC

```
<CODE BEGINS>

<t>The <strong>Killer Rabbit of Caerbannog</strong>, that most
formidable foe of
the Knights and of all that is holy or carrot-like, has been
depicted diversely in lay and in song. We venture to say,
<em>contra</em> the claim made in <relref section= "4.1" displayFormat=
"of" target="RFC8140">Ze Vompyre</relref>,
that the Killer Rabbit of Caerbannog truly is the most afeared
of all the creatures. Short of sanctified ordnance such as
<xref format= "title" target= "holy_hand_grenade"/>, there are few
remedies
known against its awful lapine powers.</t>

<CODE ENDS>
```

Figure 57: Cross-References With Attributes In AsciiRFC (RFC XML v3)

14.3. Indexing

Inline index entries are notated as "((...))". Index entries which do not appear in the text are notated as "(((...)))"; such entries may include a separate sub-entry, separated from the main entry by comma.

```
<CODE BEGINS>

The solution to the impasse at the ((Cave of Caerbannog)) was
provided by the successful deployment of the
*Holy Hand Grenade of Antioch* (see <<hand-grenade-figure>>)
(((Holy Hand Grenade of Antioch))).
Any similarity between the Holy Hand Grenade of Antioch and the
mythical _Holy Spear of Antioch_ is purely intentional;
(((relics, Christian))) any similarity between the Holy Hand Grenade
of Antioch and the _Sovereign's Orb of the United Kingdom_
(see <<sovereign-orb>>) is putatively fortuitous.
(((relics, monarchic)))

<CODE ENDS>
```

Figure 58: AsciiRFC Index Entries

```
<CODE BEGINS>

<t>The solution to the impasse at the Cave of Caerbannog<iref item=
"Cave of Caerbannog"/> was
provided by the successful deployment of the
<strong>Holy Hand Grenade of Antioch</strong> (see <xref target=
"hand-grenade-figure"/>)
<iref item= "Holy Hand Grenade of Antioch"/>.
Any similarity between the Holy Hand Grenade of Antioch and the
mythical <em>Holy Spear of Antioch</em> is purely intentional;
<iref item= "relics" subitem= "Christian"/> any similarity between the
Holy Hand Grenade
of Antioch and the <em>Sovereign's Orb of the United Kingdom</em>
(see <xref target= "sovereign-orb"/>) is putatively fortuitous.
<iref item= "relics" subitem= "monarchic"/></t>

<CODE ENDS>
```

Figure 59: AsciiRFC Index Entries (RFC XML v3)

15. Inclusions

AsciiRFC inherits the Asciidoctor syntax "include" directive [Asciidoctor-Manual] to include external files in a master AsciiRFC document.

This directive is capable of sophisticated document merging, including adjusting the heading levels of the included text, selecting text within specified tags or line numbers to be included, and adjusting the indentation of code snippets in merged text.

Its basic syntax is given in Figure 60.

```
<CODE BEGINS>
include::path[
leveloffset=_offset_,
lines=_ranges_,
tag(s)=_name(s)_,
indent=_depth_
]
<CODE ENDS>
```

Figure 60: Inclusions In AsciiRFC

If a file is included in an AsciiRFC document, ensure it ends with a blank line. An inclusion that results in its final block not being delimited with a blank line from what follows can lead to unpredictable results.

16. Encoding and Entities

XML accepts the full range of characters in the world's languages through UTF-8 character encoding, and one of the motivations for the move by the IETF from plain text to RFC XML has been to allow non-ASCII characters to be included in RFCs.

However, current RFC XML v2 tools still do not support UTF-8, and alternative tooling support for UTF-8 also remains patchy. Out of an abundance of caution, the RFC XML converter uses US-ASCII for its character encoding, and renders any non-ASCII characters as HTML entities.

AsciiRFC accepts HTML entities as input, even though they are not part of the W3C XML specification. HTML entities such as " " feature in examples of RFC XML provided by the IETF. In order to prevent dependence of the XML output from extraneous entity definitions, any such entities are rendered in the XML as decimal character entities.

An example of how AsciiRFC renders non-ASCII UTF-8 characters are given in Figure 61 with the output in Figure 62.

```
<CODE BEGINS>

____
&#x2e;&#1499;&#1488;&#1503; &#1488;&#1493;&#1500;&#1497;
&#1497;&#1502;&#1510;&#1488;&#1493;
&#1492;&#1502;&#1497;&#1500;&#1497;&#1501;
&#1492;&#1488;&#1495;&#1512;&#1493;&#1504;&#1493;&#1514; &#1513;&#1500;
&#1497;&#1493;&#1505;&#1507;
.&#1502;&#1488;&#1512;&#1502;&#1514;&#1497;&#1492;
&#x2e;&#1502;&#1497; &#1488;&#1513;&#1512; &#1497;&#1492;&#1497;&#1492;
&#1488;&#1502;&#1497;&#1509; &#1493;&#1489;&#1506;&#1500;
&#1504;&#1508;&#1513; &#1496;&#1492;&#1493;&#1512;&#1492;
&#1497;&#1493;&#1499;&#1500; &#1500;&#1502;&#1510;&#1493;&#1488;
&#1488;&#1514; &#1492;&#1490;&#1489;&#1497;&#1506;
&#1492;&#1511;&#1491;&#1493;&#1513; &#1489;&#1496;&#1497;&#1512;&#1514;
.&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1492;

"Here may be found the last words of Joseph of Arimathea.
He who is valiant and pure of spirit may find the Holy Grail
in the castle of &mdash; Aaaargh."
____

<CODE ENDS>
```

Figure 61: UTF-8 Characters In AsciiRFC

```
<CODE BEGINS>

<blockquote><t>&#1499;&#1488;&#1503; &#1488;&#1493;&#1500;&#1497;
&#1497;&#1502;&#1510;&#1488;&#1493;
&#1492;&#1502;&#1497;&#1500;&#1497;&#1501;
&#1492;&#1488;&#1495;&#1512;&#1493;&#1504;&#1493;&#1514; &#1513;&#1500;
&#1497;&#1493;&#1505;&#1507;
.&#1502;&#1488;&#1512;&#1502;&#1514;&#1497;&#1492;
&#1502;&#1497; &#1488;&#1513;&#1512; &#1497;&#1492;&#1497;&#1492;
&#1488;&#1502;&#1497;&#1509; &#1493;&#1489;&#1506;&#1500;
&#1504;&#1508;&#1513; &#1496;&#1492;&#1493;&#1512;&#1492;
&#1497;&#1493;&#1499;&#1500; &#1500;&#1502;&#1510;&#1493;&#1488;
&#1488;&#1514; &#1492;&#1490;&#1489;&#1497;&#1506;
&#1492;&#1511;&#1491;&#1493;&#1513; &#1489;&#1496;&#1497;&#1512;&#1514;
.&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1492;</t>
<t>"Here may be found the last words of Joseph of Arimathea.
He who is valiant and pure of spirit may find the Holy Grail
in the castle of &#8212; Aaaargh."</t></blockquote>

<CODE ENDS>
```

Figure 62: UTF-8 Characters In AsciiRFC Rendered As RFC XML v3

Note that because initial period is a formatting character in Asciidoctor, we have had to use "." to escape the period at the end of Hebrew sentences (which appears at the start of the line, Hebrew being written Right-to-Left). Asciidoctor is not natively equipped to deal with Right-to-Left languages in its formatting parsing.

17. Bibliography

The simple encoding of bibliography syntax provided by AsciiDoc (and Asciidoctor syntax) is inadequate for the complexity of bibliographic markup required by RFC XML.

RFC documents overwhelmingly cite other RFC documents, and canonical RFC XML bibliographic entries are available at [IETF-BibXML]; so it would be inefficient to encode those entries natively in AsciiRFC, only to have them converted back to RFC XML.

The converter provides two means of incorporating bibliographies into RFC documents authored in AsciiRFC:

o using raw RFC XML; and

o assembling bibliographies in preprocessing.

In either case, the RFC XML needs to be well-formed; missing closing tags can lead to erratic behaviour in the converter.

17.1. Using Raw RFC XML

In the first method, bibliographic citations are handled like all other AsciiRFC cross-references. The bibliographic entries for normative and informative references are given in the AsciiRFC as passthrough blocks, which contain the raw RFC XML for all references; document conversion leaves the raw RFC XML in place.

This approach requires authors to maintain the normative and informative bibliographies within the document, to update them as citations are added and removed, and to sort them manually. However, if the citation is stored on the IETF's RFC XML citation libraries (see <https://xml2rfc.tools.ietf.org>), AsciiRFC will automatically replace it with an external reference to that citation. So the body of the citation XML can be left out.

For example, the AsciiRFC in Figure 63 will generate the corresponding RFC XML v3 output in Figure 64.

<CODE BEGINS>

```
[bibliography]
== Normative References
++++
<reference anchor= "RFC2119"
target="https://www.rfc-editor.org/info/rfc2119">
<front>
<title>Key words for use in RFCs to Indicate
Requirement Levels</title>
<author initials= "S." surname= "Bradner" fullname="S. Bradner">
<organization/>
</author>
<date year= "1997" month= "March"/>
</front>
<seriesInfo name= "BCP" value= "14"/>
<seriesInfo name= "RFC" value= "2119"/>
<seriesInfo name= "DOI" value= "10.17487/RFC2119"/>
</reference>
++++


[bibliography]
== Informative References
++++

<reference anchor= "grail_film">
<front>
<title>Monty Python and the Holy Grail</title>
<author initials= "G." surname= "Chapman"/>
<author initials= "J." surname= "Cleese"/>
<author initials= "E." surname= "Idle"/>
<author initials= "T." surname= "Gilliam"/>
<author initials= "T." surname= "Jones"/>
<author initials= "M." surname= "Palin"/>
<date year= "1975"/>
</front>
</reference>

<reference anchor= "RFC2635"
target="https://www.rfc-editor.org/info/rfc2635">
<front>
<title>DON'T SPEW A Set of Guidelines for Mass Unsolicited
Mailings and Postings (spam*)</title>
<author initials= "S." surname= "Hambridge" fullname="S. Hambridge">
<organization />
</author>
<author initials= "A." surname= "Lunde" fullname="A. Lunde">
<organization />
</author>
```

```
<date year= "1999" month= "June" />
</front>
<seriesInfo name= "FYI" value= "35" />
<seriesInfo name= "RFC" value= "2635" />
<seriesInfo name= "DOI" value= "10.17487/RFC2635" />
</reference>

<reference anchor= "RFC7990"
target="https://www.rfc-editor.org/info/rfc7990">
<front>
<title>RFC Format Framework</title>
<author initials= "H." surname= "Flanagan" fullname="H. Flanagan">
<organization/>
</author>
<date year= "2016" month= "December"/>
</front>
<seriesInfo name= "RFC" value= "7990"/>
<seriesInfo name= "DOI" value= "10.17487/RFC7990"/>
</reference>

<reference anchor= "RFC8140"
target="https://www.rfc-editor.org/info/rfc8140">
<front>
<title>
The Arte of ASCII: Or, An True and Accurate Representation of
an Menagerie of Thynges Fabulous and Wonderful in Ye Forme of
Character
</title>
<author initials= "A." surname= "Farrel" fullname="A. Farrel">
<organization/>
</author>
<date year= "2017" month= "April"/>
</front>
<seriesInfo name= "RFC" value= "8140"/>
<seriesInfo name= "DOI" value= "10.17487/RFC8140"/>
</reference>

<reference anchor= 'RFC8174'
target= 'https://www.rfc-editor.org/info/rfc8174'>
<front>
<title>Ambiguity of Uppercase vs Lowercase in RFC 2119 Key
Words</title>
<author initials= 'B.' surname= 'Leiba' fullname='B. Leiba'>
<organization />
</author>
<date year= '2017' month= 'May' />
<abstract><t>RFC 2119 specifies common key words that may be used
in protocol specifications.  This document aims to reduce
```

```
the ambiguity by clarifying that only UPPERCASE usage of the
key words have the defined special meanings.</t></abstract>
</front>
<seriesInfo name= 'BCP' value= '14'/>
<seriesInfo name= 'RFC' value= '8174'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC8174'/>
</reference>

++++

<CODE ENDS>
```

Figure 63: AsciiRFC Inline Bibliography

```
<CODE BEGINS>
</section>
</middle><back>
<references anchor= "_normative_references">
<name>Normative References</name>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2119.xml"
parse= "text"/>
</references>
<references anchor= "_informative_references">
<name>Informative References</name>
<reference anchor= "grail_film">
<front>
<title>Monty Python and the Holy Grail</title>
<author initials= "G." surname= "Chapman"/>
<author initials= "J." surname= "Cleese"/>
<author initials= "E." surname= "Idle"/>
<author initials= "T." surname= "Gilliam"/>
<author initials= "T." surname= "Jones"/>
<author initials= "M." surname= "Palin"/>
<date year= "1975"/>
</front>
</reference>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2635.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7990.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8140.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8174.xml"
parse= "text"/>
</references>
</back>
</rfc>
<CODE ENDS>
```

Figure 64: AsciiRFC Inline Bibliography Rendered As RFC XML v3

17.2. Preprocessing Using "asciidoctor-bibliography"

The alternative method is to use a preprocessing tool, [asciidoctor-bibliography], to import citations into the AsciiRFC document from an external file of references.

The references file consists of RFC XML reference entries, and still needs to be managed manually; however the bibliographies are assembled from that file, sorted, and inserted into the normative and informative references in preprocessing. Citations in the document itself are given as macros to be interpreted by the preprocessor; this allows them to be split into normative and informative references. (The MMark tool likewise splits reference citations into normative and informative.)

Integration with the "asciidoc-bibliography" gem proceeds as follows:

1. Create an RFC XML references file, consisting of a "<references>" element with individual "<reference>" elements inserted, as would be done for the informative and normative references normally. The references file will contain all possible references to be used in the file; the bibliography gem will select which references have actually been cited in the document.

   A. Rather than hand crafting RFC XML references for RFC documents, you should download them from an authoritative source; e.g., "http://xml.resource.org/public/rfc/bibxml/\ reference.RFC.2119.xml". Note that RFC XML references from this link contains the XML document declaration, which needs to be removed before being used in the XML bibliography.

   B. Unlike the case for RFC XML documents created manually, the references file does not recognise XML entities and will not attempt to download them during processing. Any references to "http://xml.resource.org/public/rfc/bibxml/\ " will need to be downloaded and inserted into the references file.

   C. The RFC XML in the references file will need to be appropriate to the version of RFC XML used in the main document, as usual. Note that RFC XML v2 references are forward compatible with v3; v3 contains a couple of additional elements.

2. Add to the main document header attributes referencing the references file (":bibliography-database:"), and the bibliography style (":bibliography-style: rfc-v3").

3. References to a normative reference are inserted with the macro "cite:norm[id]" instead of "<<id>>", where "id" is the anchor of the reference.

4. References to an informative reference are inserted with the macro "cite:info[id]" instead of "<<id>>", where "id" is the anchor of the reference.

5. Formatted crossreferences and "relref" crossreferences are entered by inserting the expected raw XML in the "text" attribute. Do not use the "{cite}" interpolation of the citation. For example:

   * "<<id,words>>" = "cite:norm[id, text="<xref target= 'id'>words</xref>"]"

   * "<<id,format= counter: words>>" (processed as a formatted cross-reference) = "cite:norm[id, text="<xref format='counter' target= 'id'>words</xref>"]"

   * "<<id,2.4 comma: words>>" (processed as relref) = "cite:norm[id, text="<relref displayFormat='comma' section='2.4' target= 'id'}/>"]"

   * "<<id#section2_4,2.4 comma: words>>" (processed as relref with a cross-document internal reference) = "cite:norm[id, text="<relref relative='section2_4' displayFormat='comma' section='2.4' target= 'id'/>"]"

6. Normative and Informative References are inserted in the document through a macro, which occurs where the RFC XML references would be inserted, as shown in Figure 65.

```
<CODE BEGINS>
[bibliography]
== Normative References

++++
bibliography::norm[]
++++

[bibliography]
== Informative References

++++
bibliography::info[]
++++
<CODE ENDS>
```

Figure 65: Using asciidoctor-bibliography For Bibliography Preprocessing

18. RFC XML features not supported in Asciidoctor

The following features of RFC XML v3 [RFC7991] and v2 [RFC7749] are not supported by the AsciiRFC converter, and would need to be adjusted manually after RFC XML is generated:

RFC XML element	RFC XML v3	RFC XML v2
"front/boilerplate"	Not added by the converter	Not added by the converter
"iref@primary"	N	N
"reference" (and all children)	As Raw XML	As Raw XML
"table/preamble"	Deprecated	N
"table/postamble"	Deprecated	N
"artwork@width"	Only on images	Only on images
"artwork@height"	Only on images	Only on images

19. Authoring

To author an AsciiRFC document, you should first familiarise yourself with the [Asciidoctor-Manual].

The [asciidoctor-rfc] Ruby gem source code distribution also has samples of individual RFC XML features in v2 and v3, and examples of self-standing AsciiRFC documents, along with their RFC XML renderings. (This includes round-tripped RFC XML documents.)

19.1. Using the "rfc-asciirfc-minimal" template

In addition, you can clone the sample "rfc-asciirfc-minimal" repository as a template, and populate it for your AsciiRFC document using the steps shown in Figure 66.

```
$ git clone https://github.com/riboseinc/rfc-asciirfc-minimal
```

Figure 66: Cloning The AsciiRFC Document Template

19.2. Installing AsciiRFC Backend Processors

Converting your AsciiRFC to RFC XML is a simple as installing the Asciidoctor Ruby gem "asciidoctor" (see "Installation" at [Asciidoctor]) and the "asciidoctor-rfc" gem in Ruby (through RubyGems), then running the "asciidoctor" executable on the document, specifying the "asciidoctor-rfc" gem as a library.

The necessary steps are shown in Figure 67.

```
$ gem install asciidoctor-rfc
$ asciidoctor -b rfc3 -r 'asciidoctor-rfc' foo.adoc  # RFC XML v3 output
$ asciidoctor -b rfc2 -r 'asciidoctor-rfc' foo.adoc  # RFC XML v2 output
```

Figure 67: Installing The AsciiRFC Backend Processors

19.3. Iterating AsciiRFC Content

As you author AsciiRFC content, you should iterate by running the AsciiRFC conversion frequently, to ensure that you are still generating valid XML through your markup. The converter makes an effort to ensure that its XML output is valid, and it issues warnings about likely issues; it also validates its own XML output against the RFC XML schema (of the corresponding version), and reports errors in the XML output in the format shown in Figure 68.

```
V3 RELAXNG Validation: 12:0: ERROR: Invalid attribute
sortRefs for element rfc
```

Figure 68: Sample Validation Error Message From AsciiRFC

Note that validation against the Relax NG RFC XML schema includes confirming the referential integrity of all cross-references in the document.

It may be necessary to intervene in the XML output generated by the converter, either because the block model of AsciiRFC does not conform with the intended RFC XML (e.g. lists embedded in

paragraphs), or because RFC XML features are required that are not supported within AsciiRFC.

20. Security Considerations

- Ensure your AsciiRFC generator comes from a geniune and trustworthy source. This protects your own machine and also prevents injection of malicious content in your resulting document.

- An AsciiRFC generator may cause errors in textual rendering or link generation that may lead to security issues.

- Creating cross-references (and also bibliographic references) to external documents may pose risks since the specified external location may become controlled by a malicious party.

- URIs that start with the "http:" or "https:" prefix are automatically converted into links in AsciiRFC except when escaped with a backslash before the prefix. A URI that is intended to be text but unintentionally rendered as a link may cause users to visit a malicious website with security consequences.

- AsciiRFC contains syntax that can directly incorporate remote URI content, such as "include::" which allows remotely-located AsciiRFC content files. If a remote URI contains malicious content, this content will be included in the resulting AsciiRFC document compiled as RFC XML. Remotely-linked URIs should always be checked for malicious content prior to compiling AsciiRFC into RFC XML.

21. IANA Considerations

This document does not require any action by IANA.

22. References

22.1. Normative References

[RFC7991] Hoffman, P., "The "xml2rfc" Version 3 Vocabulary", RFC 7991, DOI 10.17487/RFC7991, December 2016, <https://www.rfc-editor.org/info/rfc7991>.

22.2. Informative References

[AsciiDoc] Rackham, S., "AsciiDoc: Text based document generation", November 2013, <http://www.methods.co.nz/asciidoc/>.

[Asciidoctor]
Allen, D., Waldron, R., and S. White, "Asciidoctor: A fast text processor & publishing toolchain for converting AsciiDoc to HTML5, DocBook & more.", November 2017, <http://asciidoctor.org>.

[asciidoctor-bibliography]
Ribose Inc., "Citations and Bibliography the 'asciidoctor-way'", November 2017, <https://github.com/riboseinc/asciidoctor-bibliography/>.

[Asciidoctor-Manual]
Allen, D., Waldron, R., and S. White, "Asciidoctor: A fast text processor & publishing toolchain for converting AsciiDoc to HTML5, DocBook & more.", November 2017, <http://asciidoctor.org/docs/user-manual/>.

[asciidoctor-rfc]
Ribose Inc., "asciidoctor-rfc lets you write Internet-Drafts and RFCs in AsciiDoc, the "asciidoctor-way".", November 2017, <https://github.com/riboseinc/asciidoctor-rfc/>.

[AsciiMathML]
"AsciiMath is an easy-to-write markup language for mathematics.", November 2017, <http://asciimath.org>.

[datatracker-asciirfc-minimal]
Carberry, J. and T. Grayson, "IETF Datatracker: A Minimal Internet-Draft In AsciiRFC", April 2018, <https://datatracker.ietf.org/doc/draft-asciirfc-minimal/>.

[datatracker-camelot-holy-grenade]
Pendragon, A., "IETF Datatracker: The Holy Hand Grenade of Antioch", Aprilt 2018, <https://datatracker.ietf.org/doc/draft-camelot-holy-grenade/>.

[datatracker-divination-cfapi]
Destiny, G. and C. Luck, "IETF Datatracker: An API For Calendar-Based Fortune Heuristics Services", March 2018, <https://datatracker.ietf.org/doc/draft-divination-cfapi/>.

[draftr] Barnes, R., "draftr: an HTML front-end to pandoc2rfc", Nov 2017, <https://ipv.sx/draftr/>.

[git-asciirfc-minimal]
Carberry, J. and T. Grayson, "Git repository: A Minimal Internet-Draft In AsciiRFC", March 2018, <https://github.com/riboseinc/rfc-asciirfc-minimal>.

[git-camelot-holy-grenade]
Pendragon, A., "Git repository: The Holy Hand Grenade of Antioch", March 2018, <https://github.com/riboseinc/rfc-camelot-holy-grenade>.

[git-divination-cfapi]
Destiny, G. and C. Luck, "Git repository: An API For Calendar-Based Fortune Heuristics Services", March 2018, <https://github.com/riboseinc/rfc-divination-cfapi>.

[I-D.asciirfc-minimal]
Carberry, J. and T. Grayson, "A Minimal Internet-Draft In AsciiRFC", draft-asciirfc-minimal-02 (work in progress), April 2018.

[I-D.camelot-holy-grenade]
Pendragon, A., "The Holy Hand Grenade of Antioch", draft-camelot-holy-grenade-01 (work in progress), April 2018.

[I-D.divination-cfapi]
Destiny, G. and C. Luck, "An API For Calendar-Based Fortune Heuristics Services", draft-divination-cfapi-00 (work in progress), March 2018.

[I-D.wu-netmod-yang-xml-doc-conventions]
Wu, Q., Farrel, A., and B. Claise, "Documentation Conventions for Expressing YANG in XML", draft-wu-netmod-yang-xml-doc-conventions-00 (work in progress), January 2018.

[IETF-BibXML]
"IETF BibXML Library", November 2017, <http://xml.resource.org/public/rfc/bibxml/\ >.

[kramdown-rfc2629]
Bormann, C., "kramdown-rfc2629: An RFC2629 (XML2RFC) backend for Thomas Leitner's kramdown markdown parser", Nov 2017, <https://github.com/cabo/kramdown-rfc2629>.

[lyx2rfc] Williams, N., "LyX to I-D/RFC export by way of Lyx export to XHTML and XSLT conversion to xml2rfc schema", 2014, <https://github.com/nicowilliams/lyx2rfc>.

[MathJax] "MathJax: A JavaScript display engine for mathematics that works in all browsers.", November 2017, <https://www.mathjax.org>.

[mmark] Gieben, R., "Using mmark to create I-Ds and RFCs", June 2015, <https://github.com/miekg/mmark>.

[NroffEdit] Santesson, S., "WYSIWYG Internet-Draft Nroff Editor", May 2011, <http://aaa-sec.com/nroffedit/>.

[pandoc2rfc] Gieben, R., "pandoc2rfc: Use pandoc to create XML suitable for xml2rfc", 2012, <https://github.com/miekg/pandoc2rfc>.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/info/rfc2119>.

[RFC5385] Touch, J., "Version 2.0 Microsoft Word Template for Creating Internet Drafts and RFCs", RFC 5385, DOI 10.17487/RFC5385, February 2010, <https://www.rfc-editor.org/info/rfc5385>.

[RFC7328] Gieben, R., "Writing I-Ds and RFCs Using Pandoc and a Bit of XML", RFC 7328, DOI 10.17487/RFC7328, August 2014, <https://www.rfc-editor.org/info/rfc7328>.

[RFC7749] Reschke, J., "The "xml2rfc" Version 2 Vocabulary", RFC 7749, DOI 10.17487/RFC7749, February 2016, <https://www.rfc-editor.org/info/rfc7749>.

[RFC7763] Leonard, S., "The text/markdown Media Type", RFC 7763, DOI 10.17487/RFC7763, March 2016, <https://www.rfc-editor.org/info/rfc7763>.

[RFC7764] Leonard, S., "Guidance on Markdown: Design Philosophies, Stability Strategies, and Select Registrations", RFC 7764, DOI 10.17487/RFC7764, March 2016, <https://www.rfc-editor.org/info/rfc7764>.

[RFC7990] Flanagan, H., "RFC Format Framework", RFC 7990, DOI 10.17487/RFC7990, December 2016, <https://www.rfc-editor.org/info/rfc7990>.

[RFC7996] Brownlee, N., "SVG Drawings for RFCs: SVG 1.2 RFC", RFC 7996, DOI 10.17487/RFC7996, December 2016, <https://www.rfc-editor.org/info/rfc7996>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/info/rfc8174>.

[TeX-LaTeX] "LaTeX is document preparation software that runs on top of Donald E. Knuth's TeX typesetting system.", November 2017, <https://www.latex-project.org>.

Appendix A. Examples

A.1. Example 1: "A Minimal Internet-Draft In AsciiRFC"

This example is available in the following formats:

- AsciiRFC [git-asciirfc-minimal]
- Internet-Draft [I-D.asciirfc-minimal]
- Text, RFC XML, PDF and HTML on the IETF Datatracker [datatracker-asciirfc-minimal]

A.1.1. In AsciiRFC

```
<CODE BEGINS>
= A Minimal Internet-Draft In AsciiRFC
:doctype: internet-draft
:name: draft-asciirfc-minimal-02
:abbrev: AsciiRFC Example
:status: informational
:ipr: trust200902
:submissionType: individual
:area: Internet
:intended-series: full-standard
:revdate: 2018-04-12T00:00:00Z
:fullname: Josiah Stinkney Carberry
:lastname: Carberry
:forename_initials: J. S.
:organization: Brown University
:phone: +1 401 863 1000
:street: Box K, 69 Brown Street
:city: Providence
:code: 02912
:country: United States of America
```

```
:uri: https://www.brown.edu
:email: josiah.carberry@ribose.com
:fullname_2: Truman Grayson
:lastname_2: Grayson
:forename_initials_2: T.
:organization_2: Brown University
:phone_2: +1 401 863 1000
:street_2: Box G, 69 Brown Street
:city_2: Providence
:code_2: 02912
:country_2: United States of America
:uri_2: https://www.brown.edu
:email_2: truman.grayson@ribose.com

[abstract]

This document provides a template on how to author (or migrate!)
a new Internet-Draft / RFC in the AsciiRFC format.

This template requires usage of the `asciidoctor-rfc` Ruby gem.

[#introduction]
== Introduction

AsciiRFC <<I-D.ribose-asciirfc>> is an extremely simple way to
author Internet-Drafts and RFCs without needing to manually
craft RFC XML conforming to <<RFC7991>>.

This is a template specifically made for authors to easily
start with creating an Internet-Draft conforming to <<RFC7991>>
and submittable to the IETF datatracker.

[#conventions]
== Terms and Definitions

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*",
"*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*",
"*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this
document are to be interpreted as described in BCP 14
<<RFC2119>> <<RFC8174>> when, and only when, they appear in
all capitals, as shown here.

This document also refers to the following terms and
definitions:

AsciiRFC::
an AsciiDoc-derived syntax used for authoring RFCs and
Internet-Drafts, as defined in <<I-D.ribose-asciirfc>>.
```

```
[#symbols]
== Symbols And Abbreviations

ADRFC::
abbreviated form of AsciiRFC


[#main]
== Main content

This is where you place the main content, and the following
serves as a placeholder for your text.

Subsections are used here for demonstration purposes.

=== Getting started

The AsciiRFC and RFC toolchains *MUST* be available locally to
build this document template.

==== AsciiRFC toolchain

You will need to have:

* Ruby: for running the AsciiRFC toolchain
* `asciidoctor-rfc` gem: for converting AsciiRFC into XML RFC
(v2 or v3)

==== XML RFC toolchain

You will need to have:

* Python: for running `xml2rfc`
* `xml2rfc`: for converting RFC XML (v2 or v3) into TXT
* `idnits`: for submission preflight


=== Referencing external content

* This is a published RFC <<RFC7253>>

* This is an Internet-Draft <<I-D.ribose-asciirfc>>

* This is an external reference <<RNP>>


[#code-snippets]
=== Code snippets
```

```
Code snippets should be wrapped with `<CODE BEGINS>` and
`<CODE ENDS>` blocks, as required by the IETF Trust Legal
Provisions (TLP) <<IETF.TLP>> specified in <<RFC5378>>.

[#security]
== Security Considerations

Any security considerations should be placed here.

As described in <<main>> (here's how you refer a local anchor),
local tools have to be installed before the document template
can be built.

Running of these local tools *MAY* produce unintended side
effects that impact security.

[#iana]
== IANA Considerations

This document does not require any action by IANA.

But if it does, such as proposing changes to IANA registries,
please include them here.

// References must be given before appendixes

[bibliography]
== Normative References

//bibliography::norm[]
++++

<reference anchor= 'RFC2119'
target= 'https://www.rfc-editor.org/info/rfc2119'>
<front>
<title>Key words for use in RFCs to Indicate Requirement
Levels</title>
<author initials= 'S.' surname= 'Bradner' fullname='S. Bradner'>
<organization />
</author>
<date year= '1997' month= 'March' />
<abstract>
<t>In many standards track documents several words are
used to signify the requirements in the specification.
These words are often capitalized. This document defines
these words as they should be interpreted in IETF
documents.  This document specifies an Internet Best
Current Practices for the Internet Community, and
```

```
  requests discussion and suggestions for improvements.
  </t>
  </abstract>
  </front>
  <seriesInfo name= 'BCP' value= '14'/>
  <seriesInfo name= 'RFC' value= '2119'/>
  <seriesInfo name= 'DOI' value= '10.17487/RFC2119'/>
  </reference>

  <reference anchor= 'RFC7991'
  target= 'https://www.rfc-editor.org/info/rfc7991'>
  <front>
  <title>The &quot;xml2rfc&quot; Version 3 Vocabulary</title>
  <author initials= 'P.' surname= 'Hoffman' fullname='P. Hoffman'>
  <organization />
  </author>
  <date year= '2016' month= 'December' />
  <abstract>
  <t>This document defines the &quot;xml2rfc&quot;
  version 3 vocabulary: an XML-based language used for
  writing RFCs and Internet-Drafts.  It is heavily derived
  from the version 2 vocabulary that is also under
  discussion.  This document obsoletes the v2 grammar
  described in RFC 7749.</t>
  </abstract>
  </front>
  <seriesInfo name= 'RFC' value= '7991'/>
  <seriesInfo name= 'DOI' value= '10.17487/RFC7991'/>
  </reference>

  <reference anchor= 'RFC8174'
  target= 'https://www.rfc-editor.org/info/rfc8174'>
  <front>
  <title>Ambiguity of Uppercase vs Lowercase in RFC 2119
  Key Words</title>
  <author initials= 'B.' surname= 'Leiba' fullname='B. Leiba'>
  <organization />
  </author>
  <date year= '2017' month= 'May' />
  <abstract>
  <t>RFC 2119 specifies common key words that may be
  used in protocol specifications.  This document aims to
  reduce the ambiguity by clarifying that only UPPERCASE
  usage of the key words have the defined special
  meanings.</t>
  </abstract>
  </front>
  <seriesInfo name= 'BCP' value= '14'/>
```

```
<seriesInfo name= 'RFC' value= '8174'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC8174'/>
</reference>
++++

[bibliography]
== Informative References

//bibliography::info[]
++++

<reference anchor= 'IETF.TLP'
target= 'https://trustee.ietf.org/trust-legal-provisions.html'>
<front>
<title>IETF Trust Legal Provisions (TLP)</title>
<author>
<organization>IETF</organization>
</author>
<date month= 'April' day= '12' year='2018' />
</front>
</reference>

<reference anchor= 'RNP' target= 'https://github.com/riboseinc/rnp/'>
<front>
<title>RNP: A C library approach to OpenPGP</title>
<author>
<organization>Ribose Inc.</organization>
<address>
<postal>
<street>Suite 1111, 1 Pedder Street</street>
<city>Central</city>
<region>Hong Kong</city>
<country>Hong Kong</country>
</postal>
<email>open.source@ribose.com</email>
<uri>https://www.ribose.com</uri>
</address>
</author>
<date day= '31' month= 'March' year='2018'/>
</front>
</reference>

<reference anchor= 'I-D.ribose-asciirfc'>
<front>
<title>
AsciiRFC: Authoring Internet-Drafts And RFCs Using AsciiDoc
</title>
<author initials= "R" surname= "Tse" fullname="Ronald Tse">
```

```
<organization/>
</author>
<author initials= "J" surname= "Lau" fullname="Jeffrey Lau">
<organization/>
</author>
<author initials= "N" surname= "Nicholas" fullname="Nick Nicholas">
<organization/>
</author>
<author initials= "P" surname= "Brasolin" fullname="Paolo Brasolin">
<organization/>
</author>
<date month= "March" day= "23" year="2018"/>
<abstract>
<t>This document describes an AsciiDoc syntax
extension called AsciiRFC, designed for authoring IETF
Internet-Drafts and RFCs.  AsciiDoc is a human readable document
markup language which affords more granular control over markup
than comparable schemes such as Markdown.  The AsciiRFC syntax
is designed to allow the author to entirely focus on text,
providing the full power of the resulting RFC XML through the
AsciiDoc language, while abstracting away the need to manually
edit XML, including references.  This document itself was
written and generated into RFC XML v2 (RFC7749) and RFC XML v3
(RFC7991) directly through asciidoctor-rfc, an AsciiRFC
generator.</t>
</abstract>
</front>
<seriesInfo name= "Internet-Draft" value= "draft-ribose-asciirfc-04"/>
<format type= "TXT" target=
"http://www.ietf.org/internet-drafts/draft-ribose-asciirfc-04.txt"/>
</reference>

<reference anchor= "RFC5378"
target="https://www.rfc-editor.org/info/rfc5378">
<front>
<title>Rights Contributors Provide to the IETF Trust</title>
<author initials= "S."
surname="Bradner" fullname="S. Bradner" role="editor">
<organization/>
</author>
<author initials= "J."
surname="Contreras" fullname="J. Contreras" role="editor">
<organization/>
</author>
<date year= "2008" month= "November"/>
<abstract><t>The IETF policies about rights in Contributions
to the IETF are designed to ensure that such Contributions
can be made available to the IETF and Internet communities
```

```
while permitting the authors to retain as many rights as
possible. This memo details the IETF policies on rights in
Contributions to the IETF. It also describes the
objectives that the policies are designed to meet. This
memo obsoletes RFCs 3978 and 4748 and, with BCP 79 and
RFC 5377, replaces Section 10 of RFC 2026. This document
specifies an Internet Best Current Practices for the
Internet Community, and requests discussion and
suggestions for improvements.</t></abstract>
</front>
<seriesInfo name= "BCP" value= "78"/>
<seriesInfo name= "RFC" value= "5378"/>
<seriesInfo name= "DOI" value= "10.17487/RFC5378"/>
</reference>

<reference anchor= 'RFC7253'
target= 'https://www.rfc-editor.org/info/rfc7253'>
<front>
<title>The OCB Authenticated-Encryption Algorithm</title>
<author initials= 'T.' surname= 'Krovetz' fullname='T. Krovetz'>
<organization />
</author>
<author initials= 'P.' surname= 'Rogaway' fullname='P. Rogaway'>
<organization />
</author>
<date year= '2014' month= 'May' />
<abstract><t>This document specifies OCB, a shared-key
blockcipher-based encryption scheme that provides
confidentiality and authenticity for plaintexts and
authenticity for associated data.  This document is a product
of the Crypto Forum Research Group (CFRG).</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '7253'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC7253'/>
</reference>
++++


[appendix]
[#appendix-a]
== Examples

=== Example 1

Here's an example of a properly wrapped code snippet in
accordance with rules specified in <<code-snippets>>.

[source,json]
```

```
  ----
  <CODE BEGINS>
  {
  "code": {
  "encoding": "ascii",
  "type":     "rfc",
  "authors":  [ "Josiah Carberry", "Truman Grayson" ]
  }
  }
  <CODE ENDS>
  ----

  [#acknowledgements]
  == Acknowledgements

  The authors would like to thank their families.

  <CODE ENDS>
```

A.1.2. Rendered as RFC XML v3

```
<CODE BEGINS>
<?xml version= "1.0" encoding= "US-ASCII"?>
<?xml-stylesheet type= "text/xsl" href= "rfc2629.xslt"?>
<!DOCTYPE rfc SYSTEM "rfc2629.dtd">
<?rfc strict= "yes"?>
<?rfc compact= "yes"?>
<?rfc subcompact= "no"?>
<?rfc toc= "yes"?>
<?rfc tocdepth= "4"?>
<?rfc symrefs= "yes"?>
<?rfc sortrefs= "yes"?>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
submissionType="IETF" prepTime="2018-04-18T03:35:29Z" version="3">
<front>
<title abbrev= "AsciiRFC Example">A Minimal Internet-Draft In
AsciiRFC</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="IETF" value="draft-asciirfc-minimal-02"/>
<seriesInfo name= "" status="full-standard"
value="draft-asciirfc-minimal-02"/>
<author fullname= "Josiah Stinkney Carberry" surname= "Carberry"
initials="J. S.">
<organization>Brown University</organization>
<address>
<postal>
<street>Box K, 69 Brown Street</street>
<city>Providence</city>
```

```
<code>02912</code>
<country>United States of America</country>
</postal>
<phone>+1 401 863 1000</phone>
<email>josiah.carberry@ribose.com</email>
<uri>https://www.brown.edu</uri>
</address>
</author>
<author fullname= "Truman Grayson" surname= "Grayson" initials="T.">
<organization>Brown University</organization>
<address>
<postal>
<street>Box G, 69 Brown Street</street>
<city>Providence</city>
<code>02912</code>
<country>United States of America</country>
</postal>
<phone>+1 401 863 1000</phone>
<email>truman.grayson@ribose.com</email>
<uri>https://www.brown.edu</uri>
</address>
</author>
<date day= "12" month= "April" year="2018"/>
<area>Internet</area>

<abstract><t>This document provides a template on how to author (or
migrate!)
a new Internet-Draft / RFC in the AsciiRFC format.</t>
<t>This template requires usage of the <tt>asciidoctor-rfc</tt> Ruby
gem.</t></abstract>
</front><middle>
<section anchor= "introduction" numbered=
"false"><name>Introduction</name><t>AsciiRFC <xref target=
"I-D.ribose-asciirfc"/> is an extremely simple way to
author Internet-Drafts and RFCs without needing to manually
craft RFC XML conforming to <xref target= "RFC7991"/>.</t>
<t>This is a template specifically made for authors to easily
start with creating an Internet-Draft conforming to <xref target=
"RFC7991"/>
and submittable to the IETF datatracker.</t></section>
<section anchor= "conventions" numbered= "false"><name>Terms and
Definitions</name><t>The key words "<bcp14>MUST</bcp14>", "<bcp14>MUST
NOT</bcp14>", "<bcp14>REQUIRED</bcp14>", "<bcp14>SHALL</bcp14>",
"<bcp14>SHALL NOT</bcp14>", "<bcp14>SHOULD</bcp14>", "<bcp14>SHOULD
NOT</bcp14>", "<bcp14>RECOMMENDED</bcp14>",
"<strong>NOT RECOMMENDED</strong>", "<bcp14>MAY</bcp14>", and
"<bcp14>OPTIONAL</bcp14>" in this
document are to be interpreted as described in BCP 14
```

```
<xref target= "RFC2119"/> <xref target="RFC8174"/> when, and only when,
they appear in
all capitals, as shown here.</t>
<t>This document also refers to the following terms and
definitions:</t>
<dl>
<dt>AsciiRFC</dt>
<dd>an AsciiDoc-derived syntax used for authoring RFCs and
Internet-Drafts, as defined in <xref target=
"I-D.ribose-asciirfc"/>.</dd>
</dl></section>
<section anchor= "symbols" numbered= "false">
<name>Symbols And Abbreviations</name>
<dl>
<dt>ADRFC</dt>
<dd>abbreviated form of AsciiRFC</dd>
</dl>
</section>
<section anchor= "main" numbered= "false"><name>Main
content</name><t>This is where you place the main content, and the
following
serves as a placeholder for your text.</t>
<t>Subsections are used here for demonstration purposes.</t>
<section anchor= "_getting_started" numbered= "false"><name>Getting
started</name><t>The AsciiRFC and RFC toolchains <bcp14>MUST</bcp14> be
available locally to
build this document template.</t>
<section anchor= "_asciirfc_toolchain" numbered= "false"><name>AsciiRFC
toolchain</name><t>You will need to have:</t>
<ul>
<li>Ruby: for running the AsciiRFC toolchain</li>
<li><tt>asciidoctor-rfc</tt> gem: for converting AsciiRFC into XML RFC
(v2 or v3)</li>
</ul></section>
<section anchor= "_xml_rfc_toolchain" numbered= "false"><name>XML RFC
toolchain</name><t>You will need to have:</t>
<ul>
<li>Python: for running <tt>xml2rfc</tt></li>
<li><tt>xml2rfc</tt>: for converting RFC XML (v2 or v3) into TXT</li>
<li><tt>idnits</tt>: for submission preflight</li>
</ul></section></section>
<section anchor= "_referencing_external_content" numbered= "false">
<name>Referencing external content</name>
<ul>
<li>This is a published RFC <xref target= "RFC7253"/></li>
<li>This is an Internet-Draft <xref target=
"I-D.ribose-asciirfc"/></li>
<li>This is an external reference <xref target= "RNP"/></li>
```

```
</ul>
</section>
<section anchor= "code-snippets" numbered= "false">
<name>Code snippets</name>
<t>Code snippets should be wrapped with <tt>&lt;CODE BEGINS&gt;</tt>
and
<tt>&lt;CODE ENDS&gt;</tt> blocks, as required by the IETF Trust Legal
Provisions (TLP) <xref target= "IETF.TLP"/> specified in <xref
target="RFC5378"/>.</t>
</section></section>
<section anchor= "security" numbered= "false"><name>Security
Considerations</name><t>Any security considerations should be placed
here.</t>
<t>As described in <xref target= "main"/> (here&#8217;s how you refer a
local anchor),
local tools have to be installed before the document template
can be built.</t>
<t>Running of these local tools <bcp14>MAY</bcp14> produce unintended
side
effects that impact security.</t></section>
<section anchor= "iana" numbered= "false"><name>IANA
Considerations</name><t>This document does not require any action by
IANA.</t>
<t>But if it does, such as proposing changes to IANA registries,
please include them here.</t></section>
</middle><back>
<references anchor= "_normative_references">
<name>Normative References</name>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2119.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7991.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8174.xml"
parse= "text"/>
</references>
<references anchor= "_informative_references">
<name>Informative References</name>
<reference anchor= "IETF.TLP" target=
"https://trustee.ietf.org/trust-legal-provisions.html">
<front>
<title>IETF Trust Legal Provisions (TLP)</title>
<author>
```

```
<organization>IETF</organization>
</author>
<date month= "April" day= "12" year="2018"/>
</front>
</reference>
<reference anchor= "RNP" target= "https://github.com/riboseinc/rnp/">
<front>
<title>RNP: A C library approach to OpenPGP</title>
<author>
<organization>Ribose Inc.</organization>
<address>
<postal>
<street>Suite 1111, 1 Pedder Street</street>
<city>Central</city>
<region>Hong Kong</region>
<country>Hong Kong</country>
</postal>
<email>open.source@ribose.com</email>
<uri>https://www.ribose.com</uri>
</address>
</author>
<date day= "31" month= "March" year="2018"/>
</front>
</reference>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml3/\
/reference.I-D.draft-ribose-asciirfc-04.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.5378.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7253.xml"
parse= "text"/>
</references>
<section anchor= "appendix-a" numbered= "false">
<name>Examples</name>
<section anchor= "_example_1" numbered= "false"><name>Example
1</name><t>Here&#8217;s an example of a properly wrapped code snippet in
accordance with rules specified in <xref target= "code-snippets"/>.</t>
<figure>
<sourcecode type= "json"><![CDATA[
<CODE BEGINS>
{
"code": {
"encoding": "ascii",
```

```
"type":     "rfc",
"authors":  [ "Josiah Carberry", "Truman Grayson" ]
}
}
<CODE ENDS>
]]></sourcecode>
</figure></section>
</section>
<section anchor= "acknowledgements" numbered= "false">
<name>Acknowledgements</name>
<t>The authors would like to thank their families.</t>
</section>
</back>
</rfc>
<CODE ENDS>
```

A.2. Example 2: "The Holy Hand Grenade of Antioch"

This example is available in the following formats:

- AsciiRFC [git-camelot-holy-grenade]

- Internet-Draft [I-D.camelot-holy-grenade]

- Text, RFC XML, PDF and HTML on the IETF Datatracker [datatracker-camelot-holy-grenade]

A.2.1. In AsciiRFC

```
<CODE BEGINS>
= The Holy Hand Grenade of Antioch
Arthur son of Uther Pendragon
:doctype: internet-draft
:abbrev: Hand Grenade of Antioch
:updates: 8140
:submission-type: independent
:name: draft-camelot-holy-grenade-01
:status: informational
:consensus: false
:area: General, Operations and Management
:keyword: rabbits, grenades, antioch, camelot
:ipr: trust200902
:toc-include: true
:sort-refs: true
:revdate: 2018-04-15T00:00:00Z
:fullname: Arthur son of Uther Pendragon
:forename_initials: A.
:lastname: Pendragon
```

```
:email: arthur.pendragon@ribose.com
:organization: Camelot
:uri: http://camelot.gov.example
:street: Palace\ Camel Lot 1
:city: Camelot
:region: England
:country: United Kingdom
:comments: yes
:notedraftinprogress: yes
:smart-quotes: false

[.comment]
tag::preamble1[]
// tag::preamble[]

[abstract]
The menagerie of beasts and artefacts depicted in RFC8140
may be usefully supplemented by other renowned figures of
Internet and more general lore. This document extends the
menagerie to the seminal fable of the
"Holy Hand Grenade of Antioch", as depicted in the
Monty Python film "Monty Python and the Holy Grail",
as well as "Spamalot", the musical inspired by the movie.

[NOTE,remove-in-rfc=false]
.Spamalot
The relevance of the musical "Spamalot" to Internet lore should be
obvious to the reader; but in case of doubt, see also
Section 1 ("What is Spam*?") of RFC2635.

// end::preamble[]
[.comment]
end::preamble1[]

[.comment]
tag::sectnums1[]
// tag::sectnums[]

[toc=exclude]
:sectnums!:
== Terminology

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*",
"*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*",
"*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this document
are to be interpreted as described in BCP 14 <<RFC2119>> <<RFC8174>>
when, and only when, they appear in all capitals, as shown here.
```

```
:sectnums:
== Introduction

<<RFC8140>> refers to the intended move of RFC formatting to
XML2RFC v3 <<RFC7990>>, in the following terms:

// end::sectnums[]
[.comment]
end::sectnums1[]

[.comment]
tag::quote1[]
// tag::quote[]

[quote,attribution="A. Farrel"]
____
Although the RFC Editor has recently dragged the IETF kicking and
screaming into the twentieth century [RFC7990] [RFC7996], there is a
yearning among all right-thinking Internet architects to "keep it
simple" and to return to the olden days when pigs could be given
thrust without anyone taking undue offence.
____

// end::quote[]
[.comment]
end::quote1[]

While no pigs, flying or otherwise, are involved in the transition
to RFC XML v3, it is opportune to enhance the <<RFC8140>>
legendarium in the service of RFC XML v3, by illustrating its
functionality through references to the mythology of Camelot, and
particularly the incidents at the Cave of Caerbannog.

[.comment]
tag::escaped_hyperlink1[]
// tag::escaped_hyperlink[]

The screaming move into the twenty-*first* century is accompanied by
a move back to the late twentieth century, with ASCII stylings more
wonted in haunts like \ftp://ftp.wwa.com/pub/Scarecrow (known to be
accessible in 1996.)

// end::escaped_hyperlink[]
[.comment]
end::escaped_hyperlink1[]

There are two references to rabbits in
_Monty Python and the Holy Grail_ which are expounded on herewith:
```

```
[.comment]
tag::listcontinuation1[]
// tag::listcontinuation[]

Trojan Rabbit::
In their siege of the French-occupied castle which may already
contain an instance of the Grail, Sir Bedevere the Wise proposes to
use a Trojan Rabbit to infiltrate the castle, with a raiding party
to take the French "not only by surprise, but totally unarmed."
+
The proposal, unsurprisingly, proved abortive. The more so as the
raiding party forgot to hide within the Trojan Rabbit, before the
French soldiers took the Trojan Rabbit inside the castle.

Killer Rabbit of Caerbannog::
Guarding the entrance to the Cave of Caerbannog; see <<caerbannog>>.

// end::listcontinuation[]
[.comment]
end::listcontinuation1[]

== The French-occupied castle

[.comment]
tag::inline_formatting1[]
// tag::inline_formatting[]

The participants of that renowned exercise in cross-cultural
communication, to wit the exchange between the
_Knights of the Round Table_
and the taunting French soldiers serving under *Guy de Lombard* are,
properly speaking, outside the scope of this `menagerie`, being more
or less human. Notwithstanding, several^ish^ beasts both animate~d~
and wooden played a significant part in this encounter; most
notably:

* The Projectile Cow, see <<projectile-cow>>
* The Trojan Rabbit, see <<trojan-rabbit>>

// end::inline_formatting[]
[.comment]
end::inline_formatting1[]


[[projectile-cow]]
.The Projectile Cow with an accompanying cannon
====
[alt=The Projectile Cow with an accompanying cannon in ASCII]
```

```
....
.-.-.-.-.-.-.-.-.-.-.-.-.--.-.-.-.-.-.-.--.-.-.-.-.-.-.-.--.-.
_-_---__--__--___-___-__-____---___-________---____-____-__-
._.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.--..-.-.-.-.-.-.--.-
,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,  ^^  .,.,.,.,  ^^   .,.,.,.=
_>-.-.-.-._>_>_>_.-.-.-.-.-.-.-.-.  \\\  .,.,.  ///  .-.-.-.-.
.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,  \ \_______/ /    .,.,.,.,
.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.  <[ {o} . ]>  #   .,.,.,.
.-.-.--.-.-.-.-.-.--.-.-.-.--.-.-.-.   [ ______]       .-.-.-.
.-.--.-.-.-.--.-.-.-.--.-.-.,.,.,.,  / [ !  ` `]   .,.,.,.,.,.-
.,.,.,.-.-,l,-,l.-,.,.,.,.-.,*.    /  {_!MOO!_}    . ., . . . ,
.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-    /M      /    -.-<>.,.,.,.-.-,
.-.-.--.-.-.-.-.-.-.-.-.--..   /MI    LK\____    .-.-.-.-.-.
.-.-.-.--.-.-.-.-.-.-.-.-.-   /MILK   mil_____k   ,.,.,.,.-,-
.-,-.-,-.,-.-,-.`-./-..     //    -`  //       .-.p . .-.-.
.-.--.-.-.-.-.-.-.-.        //   .,   //    .-.-.-.-.-.-.-.-
.-.-.--.-.-.-.-.-.-.  %____============    .-.-.--.-.-.-.-.-
-.-.-.-.--.-.-.-.-.-.      !  !           .,-.-.-,-,--,-.-,-
,--.-.-,--.--.-.,--,        \ \      .-,-,--.-,--,-.---,-.-,
,-.-.-,-,-.-,-,-.--,         +  >    .-,--,-.--,-,-.-.-,--,-
,--.-,--,-,--.---,-               .-,-,--.--,--,-.---,-,-.-.
.,.,.,.,.,.,.,.,.{A\      .,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
.,.,.,.,.,.,.{GLASS\   .,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
,.,.,.,.,.,{OF|MILK\..,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
,.,.,.,.,{ISWORTH},.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
.,.,.,.{EVERYTNG}.-.-.--.-.-.-.-.--.--.-.-.-.-.--.-.-.-.
-.-.-.-{FORINFANTS}___--___-_-__-___--*(0~`~.,.,.,.,><><.><>
_-__-_{BUTBETTER}-.-,-,-,-,-,-,-,-,.-^^^^.-.-.-.-.^^^7>>>,..
.._...{WITH_HONEY}-.-.-.-.-.-.-.-.-.RANDOM(BUSH)SHRUBS>_..
GRASS_GRASS_GRASS_GRASS_GRASS_SOMEROCKS>GRASS>GRASS<GRASS>PC
SOIL_ROOTS_SOIL_SOIL_ROCKS_SOIL_GRASS_GRASS_GRASS_ROCKS_SOIL
CLAY_ROCKS_PEBBLES_CLAY_CLAY_CLAY_CLAY_GOLD_CLAY_CLAY><_WORM
ROOTS_CLAY_SKELETON_MORESOIL_CLAY_CLAY_CLAY_CLAY_<MUSHROOMS>
....
====

[[trojan-rabbit]]
.The Trojan Rabbit with an automatic sliding door
====
[alt=The Trojan Rabbit with an automatic sliding door, in ASCII]
....
                          ___  ____
                         //_ \//\__\
                           || ||  |
                        -__||_||__|
                      //         \--_
                     //     ____     --___
                    //     //   \         \-_
```

```
                    //      \\  @/        o ||
                   //        ----      _____||
                  //                   //
             //\_//__                  //
           //--  --- \____            //
          //          --- \______   //
         //   , .          ----- \_//_
        //       ,.                --- \____
       //              .,v             --- \___
      //                                 __ -- \_
     ||  ,         _______________       //||     |-_
     ||           |   |''''''''''|     // ||     |  |
     ||     '     |   |          |        ||     |  |
     ||           |   |          |        ||     |  |
     ||      "    |   | 0        |     ___||___  |  |
     ||           |   |          |     --------  |  |
     ||___        |   |          |        ______ |  |-
    //     \      |   |          |       //     \| _| \
   //       \ ____|---|__________|______//       \/    |
  ||    X    |      /                  ||    X    |   /
   \\       /\\____/                    \\       /___/
    \\_____/ -----                       \\_____/---
     -----                                -----
....
====

[.comment]
tag::aside1[]
// tag::aside[]

****
While the exchange at the French-occupied castle is one of
the more memorable scenes of _Monty Python and the Holy Grail_,
the Trojan Rabbit has not reached the same level of cultural
resonance as its more murderous counterpart. Reasons for this
may include:

* Less overall screen-time dedicated to the Trojan Rabbit.

* The Trojan Rabbit as projectile has already been anticipated
by the Cow as projectile.
****

// end::aside[]

[.comment]
end::aside1[]
```

```
[.comment]
tag::note1[]
// tag::note[]

[NOTE,display=true,source=Author]
====
Image courtesy of
https://camelot.gov.example/creatures-in-ascii/
====

// end::note[]
[.comment]
end::note1[]


[.comment]
tag::comment1[]
// tag::comment[]

The exchange of projectile animals was the beginning of a
long-running fruitful relationship between the British and the
French peoples,
[comment]#TODO: Will need to verify that claim.# which
arguably predates the traditional English enmity with the
French. [comment]#Strictly speaking, the Knights are Welsh.#

[.comment]
--
This document, as it turns out, has a profusion of XML comments.

As expected, they are ignored in any rendering of the document.
--


// end::comment[]
[.comment]
end::comment1[]

[[caerbannog]]
== The Mythos of Caerbannog

[.comment]
tag::xref1[]
// tag::xref[]

The _Cave of Caerbannog_ has been well-established in the mythology
of Camelot (as recounted by Monty Python) as the lair of the
Legendary Black Beast of Arrrghhh, more commonly known today as the
```

```
*Killer Rabbit of Caerbannog* <<killer_rabbit_caerbannog>>.
It is the encounter between the Killer Rabbit of Caerbannog and the
Knights of the Round Table, armed with the Holy Hand Grenade of
Antioch (see the <<holy_hand_grenade,following section>>), that we
recount here through monospace font and multiple spaces.

[[killer_rabbit_caerbannog]]
=== The Killer Rabbit of Caerbannog

// end::xref[]
[.comment]
end::xref1[]

[.comment]
tag::relref1[]
// tag::relref[]

The *Killer Rabbit of Caerbannog*, that most formidable foe of
the Knights and of all that is holy or carrot-like, has been
depicted diversely in lay and in song. We venture to say,
_contra_ the claim made in <<RFC8140,4.1 of: Ze Vompyre>>,
that the Killer Rabbit of Caerbannog truly is the most afeared
of all the creatures. Short of sanctified ordnance such as
<<holy_hand_grenade,format=title>>, there are few remedies
known against its awful lapine powers.

// end::relref[]
[.comment]
end::relref1[]

[.comment]
tag::hyperlink1[]
// tag::hyperlink[]

<<killer-bunny,The following depiction>> of the fearsome beast
has been sourced from
http://camelot.gov.example/avatars/rabbit[Rabbit-SCII],
<<killer-source,accompanied>>
by C code that was used in this accurate depiction of the
Killer Rabbit:

// end::hyperlink[]
[.comment]
end::hyperlink1[]

[.comment]
tag::figure1[]
// tag::figure1a[]
```

```
[[killer-bunny]]
.A Photo Of The Killer Rabbit of Caerbannog Taken In Secret
====
[alt=The Killer Bunny, in ASCII]
....
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\^^^^^^^^^^^^^^^^^^^^^^\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\<<#MWSHARPMWMWMWTEETHWMWWM>>>\\\\\\\\\\\\\
\\\\\\\\\\\\\\\<<<#WMMWMWDEEPMDARKWCAVEMWWMMWM##>>>>\\\\\\\\
\\\\\\\\\\\\\<<#WMWMWMWMWWM/^MWMWMWMWMWMW^WMWMWMMW#>>>\\\\\\
\\\\\\\\\\\\<<#WMWMBEASTMW// \MWABBITWMW/ \MWMWMWMW##\\\\\\\
\\\\\\\\\\##MWMWMMWMWMWMWM\\  \MWMWMWMW/  /MWMWMWMWM##\\\\\\
\\\\\\\##WMWMWMWMMWMWMWMWM\\  \MWMWMW/  /MWMWMWMMWMWMWM##\\
\\\\\\##MWMMRAVENOUSMWMWMWM\\  \====/  /MWMRABBITMWMWMWMW##
\\\\\\##MWMWMWMWMMWMWMWMWM[[            ]WMWMWMMWMWMWMWMWMW
\\\\\##MWMWMWMWCARNIVOROUSW[[   3    3   ]MWMWTOOMDARKWMWMMW
\\\\##MWMWDARKMWMWMWMWMWMWM//\     o    /MWMWMWMMWMWMWMMWMWM
\\##MWMWMMKILLERABBITWMWMM//| \___vv___/ \WMPITCHWBLACKWMWMW
\##MWMWMWMMWMWMWMWMWMMWMW// |   \-^^-/   |MWMWMWMMWMWMWMWMWM
MWMWMWMMWMWVERYMDARKWMMW//  |            |MWMCAERBANNOGWMWMW
MWMWMWMMWMWMWMWMWMWMWMM{{  /             /MWMWMMWMWMWMWMWMWM
MULTRADARKWMWMHELPMWMWMW\\ \  |      |  |MWMCANMMWMWMWMMWMWW
MWMWMWMWMMWMWMWMWMMWMWMWM\\ | |_     |  |_WMWMMYOUMWMMWWMWMW
MWMMWMWMWMWMBLACKWMWMWMWM\_|__-\-----\__-\MWMWMWMREADMWMWWM
MWMWMWMMWMWMWMWMMWMWMWWMWMWMWMMWMWMWMWMWMWMWMWMWMWMMTHISWW
MWVERYMMSCARYMWMWWMWMMWMWMWMWMWMWMWMWMWMWMWMWWMWMMWMWIWM'.',
MWMWMMWMW======MWMMCANTWSEEMAMTHINGMMWMWMWMWMWMWMBETMMW` . `
MWMWMWM// SKULL \MWMWMWMMWSCREAMMMWMWMWMMWMNOTMWMWMWW  ` . \
MWMWMW|| |X||X| |MWMWCALLMMEWMMWMWMMWMWMWMWM - ` ~ . , '
MWMWMW||___ O __|MWMWMWMMWMWMWMWMMW'   ___________//   -_^_-
MWMWMW \\||_|_||MWMW      '   . .     <_|_|_||_|__|     \O/
MW   \\/\||v v||  -\\-------___     .   .,         \     |
    \\|  \_CHIN/  ==-(|CARROT/)\>     \\/||//         v\/||/
       )          /--------^-^            ,.            \|//
 #  \(/ .\\|x//                              " ' '
  . ,                \\||//        \||\\\//   \\
....
====

[[killer-source]]
.C Code To Lure Killer Rabbit Back To Cave
====
[source,c]
----
<CODE BEGINS>
/* Locate the Killer Rabbit */
int type;
```

```
unsigned char *killerRabbit =
LocateCreature(&caerbannog, "killer rabbit");
if( killerRabbit == 0 ){
puts("The Killer Rabbit of Caerbannog is out of town.");
return LOST_CREATURE;
}

/* Load Cave */
unsigned char *cave = LoadPlace(&caerbannog,
"The Cave Of Caerbannog");
if( cave == 0 ){
puts("The Cave of Caerbannog must have moved.");
return LOST_PLACE;
}

/* Lure the Killer Rabbit back into the Cave */
unsigned char *carrot = allocateObjectInPlace(
carrot("fresh"), cave);
if( carrot == 0 ){
puts("No carrot, no rabbit.");
return LOST_LURE;
}

/* Finally, notify the Killer Rabbit to act */
return notifyCreature(killerRabbit, &carrot);
<CODE ENDS>
----
====


// end::figure1a[]
[.comment]
end::figure1[]

On the beast's encounter with the Knights of the Round Table,
the following personnel engaged with it in combat:

[.comment]
tag::ul1[]
// tag::ul[]

* Killed
** Sir Bors
** Sir Gawain
** Sir Ector
* Soiled Himself
** Sir Robin
* Panicked
```

```
** King Arthur
* Employed Ordnance
** The Lector
** Brother Maynard
* Scoffed
** Tim the Enchanter

// end::ul[]
[.comment]
end::ul1[]




[[holy_hand_grenade]]
=== Holy Hand Grenade of Antioch

[.comment]
tag::figure2[]

// tag::figure2a[]

[[hand-grenade-figure]]
.The Holy Hand Grenade of Antioch (don't pull the pin)
====
[alt=Holy Hand Grenade of Antioch, in ASCII]
....
                        ______
                       \\/  \/
                      __\\  /__
                     ||  //\   |
                     ||__\\/ __|
                        ||  |    ,---,
                        ||  |====`\  |
                        ||  |    '---'
                      ,--'*`--,
                    _||#|***|#|
                  _,/.-'#|* *|#`-._
                ,,-'#####|   |#####`-.
              ,,'########|   |########`,
             //##########| o |##########\
            ||###########|   |###########|
           ||############| o |############|
           ||------------'   '------------|
           ||o  o  o  o  o   o  o  o  o  o|
            |-----------------------------|
            ||###########################|
             \\#########################/
```

```
              `..#####################,'
                ``..###############_,'
                   ``--.._______..--'
                      `''-----''`
```

```
....
====

// end::figure2a[]

[.comment]
end::figure2[]


[[sovereign-orb]]
.The Sovereign's Orb made invisible
====
.Outlines of the Sovereign's Orb
[link=https://camelot.gov.example/sovereigns_orb.jpg,align=right]
image::https://camelot.gov.example/sovereigns_orb.jpg[Orb,124,135]
====

[.comment]
tag::index1[]
// tag::index[]

The solution to the impasse at the ((Cave of Caerbannog)) was
provided by the successful deployment of the
*Holy Hand Grenade of Antioch* (see <<hand-grenade-figure>>)
(((Holy Hand Grenade of Antioch))).
Any similarity between the Holy Hand Grenade of Antioch and the
mythical _Holy Spear of Antioch_ is purely intentional;
(((relics, Christian))) any similarity between the Holy Hand Grenade
of Antioch and the _Sovereign's Orb of the United Kingdom_
(see <<sovereign-orb>>) is putatively fortuitous.
(((relics, monarchic)))

// end::index[]
[.comment]
end::index1[]

[.comment]
tag::dl1[]
// tag::dl[]

Holy Hand Grenade of Antioch::
Ordnance deployed by Brother Maynard under the incantation of a
lector, in order to dispense with the Foes of the Virtuous.
See <<hand-grenade-figure>>.
```

```
Holy Spear of Antioch::
A supposed relic of the crucifixion of Jesus Christ, this is one
of at least four claimed instances of the lance that pierced
Christ's side. Its historical significance lies in inspiring
crusaders to continue their siege of Antioch in 1098.

Sovereign's Orb of the United Kingdom::
Part of the Crown Jewels of the United Kingdom, the Sovereign's
Orb is a hollow gold sphere set with jewels and topped with a
cross.  It was made for Charles II in 1661. See <<sovereign-orb>>.

// end::dl[]
[.comment]
end::dl1[]

[.comment]
tag::bcp14_1[]
// tag::bcp14[]

The instructions in the _Book of Armaments_ on the proper deployment
of the Holy Hand Grenade of Antioch [bcp14]#may# be summarized as
follows, although this summary *SHALL NOT* be used as a substitute
for a reading from the Book of Armaments:

// end::bcp14[]
[.comment]
end::bcp14_1[]


[.comment]
tag::ol1[]
// tag::ol[]

. Preamble: St Attila Benediction
. Feast of the People on Sundry Foods
** Lambs
** Sloths
** Carp
** Anchovies
** Orangutangs
** Breakfast Cereals
** Fruit Bats
** _et hoc genus omne_
. Take out the Holy Pin
. The Count
[upperalpha]
.. Count is to Three: no more, no less
.. Not Four
```

```
.. Nor Two, except if the count then proceeds to Three
.. Five is Right Out
. Lob the Holy Hand Grenade of Antioch towards the Foe
. The Foe, being naughty in the *LORD's* sight, [bcp14]#shall# snuff it

// end::ol[]
[.comment]
end::ol1[]

This could also be represented in pseudocode as follows:

[.comment]
tag::listcontinuationblock1[]
// tag::listcontinuationblock[]

. Take out the Holy Pin
. The Count
+
----
integer count;
for count := 1 step 1 until 3 do
say(count)
comment Five is Right Out
----
. Lob the Holy Hand Grenade of Antioch towards the Foe
. Foe snuffs it

// end::listcontinuationblock[]
[.comment]
end::listcontinuationblock1[]

== Dramatis Personae

The following human (more-or-less) protagonists were involved
in the two incidents recounted as lore of the Knights of the
Round Table:

[.comment]
tag::table1[]
// tag::table[]

[grid=all,options="footer"]
|===
|French Castle | Cave of Caerbannog

2+|King Arthur
2+|Patsy
2+|Sir Bedevere the Wise
```

```
2+|Sir Galahad the Pure
2+|Sir Lancelot the Brave
2+|Sir Robin the Not-quite-so-brave-as-Sir-Lancelot
|French Guard with Outrageous Accent| Tim the Enchanter
|Other French Guards | Brother Maynard
| | The Lector
.3+^|not yet recruited
>|Sir Bors
>|Sir Gawain
>|Sir Ector

|Retinue of sundry knights
|Retinue of sundry more knights than at the French Castle
|===

// end::table[]
[.comment]
end::table1[]

=== Past the Killer Rabbit

Once the Killer Rabbit of Caerbannog (<<killer-bunny>>) had been
dispatched, the Knights of the Round Table uncovered the last
words of Joseph of Arimathea, inscribed on the Cave of Caerbannog
in Aramaic.  While the precise Aramaic wording has not survived,
we trust the following Hebrew subtitles will serve as an
acceptable substitute:

[.comment]
tag::hebrew1[]
// tag::hebrew[]

____
&#x2e;&#1499;&#1488;&#1503; &#1488;&#1493;&#1500;&#1497;
&#1497;&#1502;&#1510;&#1488;&#1493;
&#1492;&#1502;&#1497;&#1500;&#1497;&#1501;
&#1492;&#1488;&#1495;&#1512;&#1493;&#1504;&#1493;&#1514; &#1513;&#1500;
&#1497;&#1493;&#1505;&#1507;
.&#1502;&#1488;&#1512;&#1502;&#1514;&#1497;&#1492;
&#x2e;&#1502;&#1497; &#1488;&#1513;&#1512; &#1497;&#1492;&#1497;&#1492;
&#1488;&#1502;&#1497;&#1509; &#1493;&#1489;&#1506;&#1500;
&#1504;&#1508;&#1513; &#1496;&#1492;&#1493;&#1512;&#1492;
&#1497;&#1493;&#1499;&#1500; &#1500;&#1502;&#1510;&#1493;&#1488;
&#1488;&#1514; &#1492;&#1490;&#1489;&#1497;&#1506;
&#1492;&#1511;&#1491;&#1493;&#1513; &#1489;&#1496;&#1497;&#1512;&#1514;
.&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1492;

"Here may be found the last words of Joseph of Arimathea.
```

```
He who is valiant and pure of spirit may find the Holy Grail
in the castle of &mdash; Aaaargh."
____

// end::hebrew[]
[.comment]
end::hebrew1[]


== IANA Considerations

IANA might consider a registry to track the mythical, especially
ravaging beasts, such as the Killer Rabbit, who haunt the Internet.


== Security Considerations

Do not let the Killer Rabbit out under any circumstance.

I repeat. Do not let the Killer Rabbit (<<killer-bunny>>) out.


[.comment]
tag::bibliography1[]
// tag::bibliography[]

[bibliography]
== Normative References
++++
<reference anchor= "RFC2119"
target="https://www.rfc-editor.org/info/rfc2119">
<front>
<title>Key words for use in RFCs to Indicate
Requirement Levels</title>
<author initials= "S." surname= "Bradner" fullname="S. Bradner">
<organization/>
</author>
<date year= "1997" month= "March"/>
</front>
<seriesInfo name= "BCP" value= "14"/>
<seriesInfo name= "RFC" value= "2119"/>
<seriesInfo name= "DOI" value= "10.17487/RFC2119"/>
</reference>
++++


[bibliography]
== Informative References
```

```
++++

<reference anchor= "grail_film">
<front>
<title>Monty Python and the Holy Grail</title>
<author initials= "G." surname= "Chapman"/>
<author initials= "J." surname= "Cleese"/>
<author initials= "E." surname= "Idle"/>
<author initials= "T." surname= "Gilliam"/>
<author initials= "T." surname= "Jones"/>
<author initials= "M." surname= "Palin"/>
<date year= "1975"/>
</front>
</reference>

<reference anchor= "RFC2635"
target="https://www.rfc-editor.org/info/rfc2635">
<front>
<title>DON'T SPEW A Set of Guidelines for Mass Unsolicited
Mailings and Postings (spam*)</title>
<author initials= "S." surname= "Hambridge" fullname="S. Hambridge">
<organization />
</author>
<author initials= "A." surname= "Lunde" fullname="A. Lunde">
<organization />
</author>
<date year= "1999" month= "June" />
</front>
<seriesInfo name= "FYI" value= "35" />
<seriesInfo name= "RFC" value= "2635" />
<seriesInfo name= "DOI" value= "10.17487/RFC2635" />
</reference>

<reference anchor= "RFC7990"
target="https://www.rfc-editor.org/info/rfc7990">
<front>
<title>RFC Format Framework</title>
<author initials= "H." surname= "Flanagan" fullname="H. Flanagan">
<organization/>
</author>
<date year= "2016" month= "December"/>
</front>
<seriesInfo name= "RFC" value= "7990"/>
<seriesInfo name= "DOI" value= "10.17487/RFC7990"/>
</reference>

<reference anchor= "RFC8140"
target="https://www.rfc-editor.org/info/rfc8140">
```

```
<front>
<title>
The Arte of ASCII: Or, An True and Accurate Representation of
an Menagerie of Thynges Fabulous and Wonderful in Ye Forme of
Character
</title>
<author initials= "A." surname= "Farrel" fullname="A. Farrel">
<organization/>
</author>
<date year= "2017" month= "April"/>
</front>
<seriesInfo name= "RFC" value= "8140"/>
<seriesInfo name= "DOI" value= "10.17487/RFC8140"/>
</reference>

<reference anchor= 'RFC8174'
target= 'https://www.rfc-editor.org/info/rfc8174'>
<front>
<title>Ambiguity of Uppercase vs Lowercase in RFC 2119 Key
Words</title>
<author initials= 'B.' surname= 'Leiba' fullname='B. Leiba'>
<organization />
</author>
<date year= '2017' month= 'May' />
<abstract><t>RFC 2119 specifies common key words that may be used
in protocol specifications.  This document aims to reduce
the ambiguity by clarifying that only UPPERCASE usage of the
key words have the defined special meanings.</t></abstract>
</front>
<seriesInfo name= 'BCP' value= '14'/>
<seriesInfo name= 'RFC' value= '8174'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC8174'/>
</reference>

++++

// end::bibliography[]
[.comment]
end::bibliography1[]
<CODE ENDS>
```

A.2.2. Rendered as RFC XML v3

```
<CODE BEGINS>
<?xml version= "1.0" encoding= "US-ASCII"?>
<?xml-stylesheet type= "text/xsl" href= "rfc2629.xslt"?>
<!DOCTYPE rfc SYSTEM "rfc2629.dtd">
<?rfc comments= "yes"?>
```

```
<?rfc notedraftinprogress= "yes"?>
<?rfc strict= "yes"?>
<?rfc compact= "yes"?>
<?rfc subcompact= "no"?>
<?rfc toc= "yes"?>
<?rfc tocdepth= "4"?>
<?rfc symrefs= "yes"?>
<?rfc sortrefs= "true"?>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
updates="8140" sortRefs="true" tocInclude="true"
submissionType="independent" prepTime="2018-04-18T03:35:33Z"
version="3">
<front>
<title abbrev= "Hand Grenade of Antioch">The Holy Hand Grenade of
Antioch</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="independent" value="draft-camelot-holy-grenade-01"/>
<author fullname= "Arthur son of Uther Pendragon" surname= "Pendragon"
initials="A.">
<organization>Camelot</organization>
<address>
<postal>
<street>Palace</street>
<street>Camel Lot 1</street>
<city>Camelot</city>
<region>England</region>
<country>United Kingdom</country>
</postal>
<email>arthur.pendragon@ribose.com</email>
<uri>http://camelot.gov.example</uri>
</address>
</author>
<date day= "15" month= "April" year="2018"/>
<area>General</area>
<area>Operations and Management</area>
<keyword>rabbits</keyword>
<keyword>grenades</keyword>
<keyword>antioch</keyword>
<keyword>camelot</keyword>

<abstract>

<!-- tag::preamble1[] -->


<t>The menagerie of beasts and artefacts depicted in RFC8140
may be usefully supplemented by other renowned figures of
Internet and more general lore. This document extends the
```

```
menagerie to the seminal fable of the
"Holy Hand Grenade of Antioch", as depicted in the
Monty Python film "Monty Python and the Holy Grail",
as well as "Spamalot", the musical inspired by the
movie.</t></abstract><note removeInRFC= "false">
<name>Spamalot</name>
<t>The relevance of the musical "Spamalot" to Internet lore should be
obvious to the reader; but in case of doubt, see also
Section 1 ("What is Spam*?") of RFC2635.</t>
</note>


<!-- end::preamble1[] -->




<!-- tag::sectnums1[] -->


</front><middle>
<section anchor= "_terminology" toc= "exclude" numbered="false">
<name>Terminology</name>
<t>The key words "<bcp14>MUST</bcp14>", "<bcp14>MUST NOT</bcp14>",
"<bcp14>REQUIRED</bcp14>", "<bcp14>SHALL</bcp14>",
"<bcp14>SHALL NOT</bcp14>", "<bcp14>SHOULD</bcp14>", "<bcp14>SHOULD
NOT</bcp14>", "<bcp14>RECOMMENDED</bcp14>",
"<strong>NOT RECOMMENDED</strong>", "<bcp14>MAY</bcp14>", and
"<bcp14>OPTIONAL</bcp14>" in this document
are to be interpreted as described in BCP 14 <xref target= "RFC2119"/>
<xref target="RFC8174"/>
when, and only when, they appear in all capitals, as shown here.</t>
</section>
<section anchor= "_introduction" numbered=
"true"><name>Introduction</name><t><xref target= "RFC8140"/> refers to
the intended move of RFC formatting to
XML2RFC v3 <xref target= "RFC7990"/>, in the following terms:</t>


<!-- end::sectnums1[] -->




<!-- tag::quote1[] -->


<blockquote quotedFrom= "A. Farrel">
```

```
<t>Although the RFC Editor has recently dragged the IETF kicking and
screaming into the twentieth century [RFC7990] [RFC7996], there is a
yearning among all right-thinking Internet architects to "keep it
simple" and to return to the olden days when pigs could be given
thrust without anyone taking undue offence.</t>
</blockquote>


<!-- end::quote1[] -->


<t>While no pigs, flying or otherwise, are involved in the transition
to RFC XML v3, it is opportune to enhance the <xref target= "RFC8140"/>
legendarium in the service of RFC XML v3, by illustrating its
functionality through references to the mythology of Camelot, and
particularly the incidents at the Cave of Caerbannog.</t>


<!-- tag::escaped_hyperlink1[] -->


<t>The screaming move into the twenty-<strong>first</strong> century is
accompanied by
a move back to the late twentieth century, with ASCII stylings more
wonted in haunts like ftp://ftp.wwa.com/pub/Scarecrow (known to be
accessible in 1996.)</t>


<!-- end::escaped_hyperlink1[] -->


<t>There are two references to rabbits in
<em>Monty Python and the Holy Grail</em> which are expounded on
herewith:</t>


<!-- tag::listcontinuation1[] -->


<dl>
<dt>Trojan Rabbit</dt>
<dd>
<t>In their siege of the French-occupied castle which may already
contain an instance of the Grail, Sir Bedevere the Wise proposes to
use a Trojan Rabbit to infiltrate the castle, with a raiding party
to take the French "not only by surprise, but totally unarmed."</t>
<t>The proposal, unsurprisingly, proved abortive. The more so as the
raiding party forgot to hide within the Trojan Rabbit, before the
```

```
French soldiers took the Trojan Rabbit inside the castle.</t>
</dd>
<dt>Killer Rabbit of Caerbannog</dt>
<dd>Guarding the entrance to the Cave of Caerbannog; see <xref target=
"caerbannog"/>.</dd>
</dl>


<!-- end::listcontinuation1[] -->

</section>
<section anchor= "_the_french_occupied_castle" numbered=
"true"><name>The French-occupied castle</name>

<!-- tag::inline_formatting1[] -->


<t>The participants of that renowned exercise in cross-cultural
communication, to wit the exchange between the
<em>Knights of the Round Table</em>
and the taunting French soldiers serving under <strong>Guy de
Lombard</strong> are,
properly speaking, outside the scope of this <tt>menagerie</tt>, being
more
or less human. Notwithstanding, several<sup>ish</sup> beasts both
animate<sub>d</sub>
and wooden played a significant part in this encounter; most
notably:</t>
<ul>
<li>The Projectile Cow, see <xref target= "projectile-cow"/></li>
<li>The Trojan Rabbit, see <xref target= "trojan-rabbit"/></li>
</ul>


<!-- end::inline_formatting1[] -->


<figure anchor= "projectile-cow">
<name>The Projectile Cow with an accompanying cannon</name>
<artwork type= "ascii-art" alt= "The Projectile Cow with an
accompanying cannon in ASCII"><![CDATA[
.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.
_-_---__--__--___-___-__-____---___-________---____-____-__-
._.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-.-
,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,  ^^  .,.,.,.,  ^^   .,.,.,.=
_>-.-.-.-._>_>_>_.-.-.-.-.-.-.-.-.  \\\  .,.,.  ///  .-.-.-.-.
.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,  \ \_______/ /    .,.,.,.,
.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.  <[ {o} . ]>  #   .,.,.,.
```

```
.-.-.--.-.-.-.-.-.--.-.-.-.--.-.-.   [ ______]       .-.-.-.
.-.--.-.-.-.--.-.-.-.--.-.-.,.,.,  / [ !  ` `]   .,.,..,.,.-
.,.,.,.-.-,l,-,l.-,.,.,.,-.,*.    /  {_!MOO!_}    . ., . . ,
.-.-.-.-.-.-.-.-.-.-.-.-.-.-    /M      /    -.-<>.,.,..-.-,
.-.-.--.-.-.-.-.-.-.-.--..   /MI    LK\____    .-.-.-.-.-.
.-.-.-.--.-.-.-.-.-.-.-.-   /MILK   mil_____k   ,.,.,..-,-
.-,-.-,-.,-.-,-.`-./-..     //    -`  //       .-.p . .-.-.
.-.--.-.-.-.-.-.-.        //   .,   //    .-.-.-.-.-.-.-.-
.-.-.--.-.-.-.-.-.  %____============    .-.-.--.-.-.-.-.-
-.-.-.-.--.-.-.-.-.      !  !           .,-.-.-,-,--,-.-,-
,--.-.-,--.--.-.,--,        \ \      .-,-,--.-,--,-.---,-.-,
,-.-.-,-,-.-,-,-.--,         +  >    .-,--,-.--,-,-.-.-,--,-
,--.-,--,-,--.---,-               .-,-,--.--,--,-.---,-,-.-.
.,.,.,.,.,.,.,.{A\      .,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
.,.,.,.,.,.,.{GLASS\   .,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
,.,.,.,.,.,{OF|MILK\..,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,
,.,.,.,.,.{ISWORTH},.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.,.
.,.,.,.,.{EVERYTNG}.-.-.--.-.-.-.-.--.--.-.-.-.-.--.-.-.-.
-.-.-.-{FORINFANTS}___--___-_-__-___--*(0~`~.,.,.,.,><><.><>
_-__-_{BUTBETTER}-.-,-,-,-,-,-,-,-,.-^^^^.-.-.-.-.^^^7>>>,..
.._...{WITH_HONEY}-.-.-.-.-.-.-.-.-.-.RANDOM(BUSH)SHRUBS>_..
GRASS_GRASS_GRASS_GRASS_GRASS_SOMEROCKS>GRASS>GRASS<GRASS>PC
SOIL_ROOTS_SOIL_SOIL_ROCKS_SOIL_GRASS_GRASS_GRASS_ROCKS_SOIL
CLAY_ROCKS_PEBBLES_CLAY_CLAY_CLAY_CLAY_GOLD_CLAY_CLAY><_WORM
ROOTS_CLAY_SKELETON_MORESOIL_CLAY_CLAY_CLAY_CLAY_<MUSHROOMS>
]]></artwork>
</figure>
<figure anchor= "trojan-rabbit">
<name>The Trojan Rabbit with an automatic sliding door</name>
<artwork type= "ascii-art" alt= "The Trojan Rabbit with an automatic
sliding door"><![CDATA[
                          ___  ____
                         //_ \//\__\
                           || ||  |
                        -__||_||__|
                      //         \--_
                     //     ____     --___
                    //     //   \         \-_
                   //      \\  @/        o ||
                  //        ----      _____||
                 //                    //
             //\_//__                 //
            //--  --- \____           //
           //          --- \______   //
          //   , .          ----- \_//_
         //       ,.                --- \____
        //              .,v             --- \___
       //                                __ -- \_
```

```
      ||  ,         _______________       //||     |-_
      ||           |   |''''''''''|     // ||     |  |
      ||     '     |   |          |        ||     |  |
      ||           |   |          |        ||     |  |
      ||      "    |   | 0        |     ___||___  |  |
      ||           |   |          |     --------  |  |
      ||___        |   |          |        ______ |  |-
     //     \      |   |          |       //     \| _| \
    //       \ ____|---|__________|______//       \/    |
   ||    X    |      /                  ||    X    |   /
    \\       /\\____/                    \\       /___/
     \\_____/ -----                       \\_____/---
      -----                                -----
]]></artwork>
</figure>


<!-- tag::aside1[] -->


<aside><t>While the exchange at the French-occupied castle is one of
the more memorable scenes of <em>Monty Python and the Holy Grail</em>,
the Trojan Rabbit has not reached the same level of cultural
resonance as its more murderous counterpart. Reasons for this
may include:</t>
<ul>
<li>Less overall screen-time dedicated to the Trojan Rabbit.</li>
<li>The Trojan Rabbit as projectile has already been anticipated
by the Cow as projectile.</li>
</ul></aside>


<!-- end::aside1[] -->




<!-- tag::note1[] -->


<t><cref display= "true" source= "Author">Image courtesy of
<eref target=
"https://camelot.gov.example/creatures-in-ascii/"/></cref></t>


<!-- end::note1[] -->
```

```
<!-- tag::comment1[] -->


<t>The exchange of projectile animals was the beginning of a
long-running fruitful relationship between the British and the
French peoples,


<!-- TODO: Will need to verify that claim. -->

which
arguably predates the traditional English enmity with the
French.

<!-- Strictly speaking, the Knights are Welsh. -->

</t>


<!-- This document, as it turns out, has a profusion of XML comments.

As expected, they are ignored in any rendering of the document.
-->




<!-- end::comment1[] -->

</section>
<section anchor= "caerbannog" numbered= "true"><name>The Mythos of
Caerbannog</name>

<!-- tag::xref1[] -->


<t>The <em>Cave of Caerbannog</em> has been well-established in the
mythology
of Camelot (as recounted by Monty Python) as the lair of the
Legendary Black Beast of Arrrghhh, more commonly known today as the
<strong>Killer Rabbit of Caerbannog</strong> <xref target=
"killer_rabbit_caerbannog"/>.
It is the encounter between the Killer Rabbit of Caerbannog and the
Knights of the Round Table, armed with the Holy Hand Grenade of
Antioch (see the <xref target= "holy_hand_grenade">following
section</xref>), that we
recount here through monospace font and multiple spaces.</t>
<section anchor= "killer_rabbit_caerbannog" numbered= "true"><name>The
```

```
Killer Rabbit of Caerbannog</name>

<!-- end::xref1[] -->



<!-- tag::relref1[] -->


<t>The <strong>Killer Rabbit of Caerbannog</strong>, that most
formidable foe of
the Knights and of all that is holy or carrot-like, has been
depicted diversely in lay and in song. We venture to say,
<em>contra</em> the claim made in <relref section= "4.1" displayFormat=
"of" target="RFC8140">Ze Vompyre</relref>,
that the Killer Rabbit of Caerbannog truly is the most afeared
of all the creatures. Short of sanctified ordnance such as
<xref format= "title" target= "holy_hand_grenade"/>, there are few
remedies
known against its awful lapine powers.</t>


<!-- end::relref1[] -->



<!-- tag::hyperlink1[] -->


<t><xref target= "killer-bunny">The following depiction</xref> of the
fearsome beast
has been sourced from
<eref target=
"http://camelot.gov.example/avatars/rabbit">Rabbit-SCII</eref>,
<xref target= "killer-source">accompanied</xref>
by C code that was used in this accurate depiction of the
Killer Rabbit:</t>


<!-- end::hyperlink1[] -->



<!-- tag::figure1[] -->
```

```
<figure anchor= "killer-bunny">
<name>A Photo Of The Killer Rabbit of Caerbannog Taken In
Secret</name>
<artwork type= "ascii-art" alt= "The Killer Bunny"><![CDATA[
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\\\\\^^^^^^^^^^^^^^^^^^^^^^^\\\\\\\\\\\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\\\\\\<<#MWSHARPMWMWMWTEETHWMWWM>>>\\\\\\\\\\\\\\
\\\\\\\\\\\\\\\\<<<#WMMWMWDEEPMDARKWCAVEMWWMMWM##>>>>\\\\\\\\
\\\\\\\\\\\\\<<#WMWMWMWMWWM/^MWMWMWMWMWMW^WMWMWMMW#>>>\\\\\\
\\\\\\\\\\\\<<#WMWMBEASTMW// \MWABBITWMW/ \MWMWMWMW##\\\\\\\
\\\\\\\\\\##MWMWMMWMWMWMWM\\  \MWMWMWMW/  /MWMWMWMWM##\\\\\\
\\\\\\\##WMWMWMWMMWMWMWMWM\\  \MWMWMW/  /MWMWMWMMWMWMWM##\\
\\\\\\##MWMMRAVENOUSMWMWMWM\\  \====/  /MWMRABBITMWMWMWMW##
\\\\\##MWMWMWMWMMWMWMWMWMW[[            ]WMWMWMMWMWMWMWMWMW
\\\\##MWMWMWMWCARNIVOROUSW[[   3    3   ]MWMWTOOMDARKWMWMMW
\\\##MWMWDARKMWMWMWMWMWMW//\     o    /MWMWMWMMWMWMWMMWMWM
\\##MWMWMMKILLERABBITWMWMM//| \___vv___/ \WMPITCHWBLACKWMWMW
\##MWMWMWMMWMWMWMWMWMMWMW// |   \-^^-/   |MWMWMWMMWMWMWMWMWM
MWMWMWMMWMWVERYMDARKWMMW//  |            |MWMCAERBANNOGWMWMW
MWMWMWMMWMWMWMWMWMWMWMM{{  /             /MWMWMMWMWMWMWMWMWM
MULTRADARKWMWMHELPMWMWMW\\ \  |      |  |MWMCANMMWMWMWMMWMWW
MWMWMWMWMMWMWMWMWMMWMWMWM\\ | |_     |  |_WMWMMYOUMWMMWWMWMW
MWMMWMWMWMWMBLACKWMWMWMWWM\_|__-\-----\__-\MWMWMWMREADMWMWWM
MWMWMWMMWMWMWMWMMWMWMWWMWMWMWMWMMWMWMWMWMWMWMWMWMWMWMMTHISWW
MWVERYMMSCARYMWMWWMWMMWMWMWMWMWMWMWMWMWMWMWMWWMWMMWMWIWM'.',
MWMWMMWMW======MWMMCANTWSEEMAMTHINGMMWMWMWMWMWMWMBETMMW` . `
MWMWMWM// SKULL \MWMWMWMMWSCREAMMMWMWMWMMWMNOTMWMWMWW  ` . \
MWMWMW|| |X||X| |MWMWCALLMMEWMMWMWMMWMWMWMWWM - ` ~ . , '
MWMWMW||___ O __|MWMWMWMMWMWMWMWMMW'   ___________//   -_^_-
MWMWMW \\||_|_||MWMW      '   . .     <_|_|_||_|__|     \O/
MW   \\/\||v v||  -\\-------___     .   .,         \     |
    \\|  \_CHIN/  ==-(|CARROT/)\>     \\/||//         v\/||/
       )          /--------^-^            ,.            \|//
 #  \(/ .\\|x//                              " ' '
  . ,               \\||//        \||\\\//   \\
]]></artwork>
</figure>
<figure anchor= "killer-source">
<name>C Code To Lure Killer Rabbit Back To Cave</name>
<sourcecode type= "c"><![CDATA[
<CODE BEGINS>
/* Locate the Killer Rabbit */
int type;
unsigned char *killerRabbit =
LocateCreature(&caerbannog, "killer rabbit");
if( killerRabbit == 0 ){
puts("The Killer Rabbit of Caerbannog is out of town.");
```

```
return LOST_CREATURE;
}

/* Load Cave */
unsigned char *cave = LoadPlace(&caerbannog,
"The Cave Of Caerbannog");
if( cave == 0 ){
puts("The Cave of Caerbannog must have moved.");
return LOST_PLACE;
}

/* Lure the Killer Rabbit back into the Cave */
unsigned char *carrot = allocateObjectInPlace(
carrot("fresh"), cave);
if( carrot == 0 ){
puts("No carrot, no rabbit.");
return LOST_LURE;
}

/* Finally, notify the Killer Rabbit to act */
return notifyCreature(killerRabbit, &carrot);
<CODE ENDS>
]]></sourcecode>
</figure>


<!-- end::figure1[] -->


<t>On the beast's encounter with the Knights of the Round Table,
the following personnel engaged with it in combat:</t>


<!-- tag::ul1[] -->


<ul>
<li>
<t>Killed</t>
<ul>
<li>Sir Bors</li>
<li>Sir Gawain</li>
<li>Sir Ector</li>
</ul>
</li>
<li>
<t>Soiled Himself</t>
<ul>
```

```
<li>Sir Robin</li>
</ul>
</li>
<li>
<t>Panicked</t>
<ul>
<li>King Arthur</li>
</ul>
</li>
<li>
<t>Employed Ordnance</t>
<ul>
<li>The Lector</li>
<li>Brother Maynard</li>
</ul>
</li>
<li>
<t>Scoffed</t>
<ul>
<li>Tim the Enchanter</li>
</ul>
</li>
</ul>


<!-- end::ul1[] -->

</section>
<section anchor= "holy_hand_grenade" numbered= "true"><name>Holy Hand
Grenade of Antioch</name>

<!-- tag::figure2[] -->


<figure anchor= "hand-grenade-figure">
<name>The Holy Hand Grenade of Antioch (don't pull the pin)</name>
<artwork type= "ascii-art" alt= "Holy Hand Grenade of
Antioch"><![CDATA[
                        ______
                       \\/  \/
                      __\\  /__
                     ||  //\   |
                     ||__\\/ __|
                        ||  |    ,---,
                        ||  |====`\  |
                        ||  |    '---'
                      ,--'*`--,
                    _||#|***|#|
```

```
                  _,/.-'#|* *|#`-._
                ,,-'#####|   |#####`-.
              ,,'########|   |########`,
             //##########| o |##########\
            ||###########|   |###########|
           ||############| o |############|
           ||------------'   '------------|
           ||o  o  o  o  o   o  o  o  o  o|
            |-----------------------------|
            ||###########################|
             \\#########################/
              `..#####################,'
               ``..###############_,'
                 ``--.._____..--'
                    `''-----''`
```

]]></artwork>
</figure>

<!-- end::figure2[] -->

<figure anchor= "sovereign-orb">
<name>The Sovereign's Orb made invisible</name>
<artwork align= "right" alt= "Orb" height="135" name="Outlines of the
Sovereign's Orb" src="https://camelot.gov.example/sovereigns_orb.jpg"
type="binary-art" width="124"/>
</figure>

<!-- tag::index1[] -->

<t>The solution to the impasse at the Cave of Caerbannog<iref item=
"Cave of Caerbannog"/> was
provided by the successful deployment of the
Holy Hand Grenade of Antioch (see <xref target=
"hand-grenade-figure"/>)
<iref item= "Holy Hand Grenade of Antioch"/>.
Any similarity between the Holy Hand Grenade of Antioch and the
mythical Holy Spear of Antioch is purely intentional;
<iref item= "relics" subitem= "Christian"/> any similarity between the
Holy Hand Grenade
of Antioch and the Sovereign's Orb of the United Kingdom
(see <xref target= "sovereign-orb"/>) is putatively fortuitous.
<iref item= "relics" subitem= "monarchic"/></t>

```
<!-- end::index1[] -->



<!-- tag::dl1[] -->


<dl>
<dt>Holy Hand Grenade of Antioch</dt>
<dd>Ordnance deployed by Brother Maynard under the incantation of a
lector, in order to dispense with the Foes of the Virtuous.
See <xref target= "hand-grenade-figure"/>.</dd>
<dt>Holy Spear of Antioch</dt>
<dd>A supposed relic of the crucifixion of Jesus Christ, this is one
of at least four claimed instances of the lance that pierced
Christ's side. Its historical significance lies in inspiring
crusaders to continue their siege of Antioch in 1098.</dd>
<dt>Sovereign's Orb of the United Kingdom</dt>
<dd>Part of the Crown Jewels of the United Kingdom, the Sovereign's
Orb is a hollow gold sphere set with jewels and topped with a
cross.  It was made for Charles II in 1661. See <xref target=
"sovereign-orb"/>.</dd>
</dl>


<!-- end::dl1[] -->



<!-- tag::bcp14_1[] -->


<t>The instructions in the <em>Book of Armaments</em> on the proper
deployment
of the Holy Hand Grenade of Antioch <bcp14>MAY</bcp14> be summarized as
follows, although this summary <bcp14>SHALL NOT</bcp14> be used as a
substitute
for a reading from the Book of Armaments:</t>


<!-- end::bcp14_1[] -->



<!-- tag::ol1[] -->
```

```
<ol type= "1">
<li>Preamble: St Attila Benediction</li>
<li>
<t>Feast of the People on Sundry Foods</t>
<ul>
<li>Lambs</li>
<li>Sloths</li>
<li>Carp</li>
<li>Anchovies</li>
<li>Orangutangs</li>
<li>Breakfast Cereals</li>
<li>Fruit Bats</li>
<li>
<em>et hoc genus omne</em>
</li>
</ul>
</li>
<li>Take out the Holy Pin</li>
<li>
<t>The Count</t>
<ol type= "A">
<li>Count is to Three: no more, no less</li>
<li>Not Four</li>
<li>Nor Two, except if the count then proceeds to Three</li>
<li>Five is Right Out</li>
</ol>
</li>
<li>Lob the Holy Hand Grenade of Antioch towards the Foe</li>
<li>The Foe, being naughty in the <strong>LORD's</strong> sight,
<bcp14>SHALL</bcp14> snuff it</li>
</ol>


<!-- end::ol1[] -->


<t>This could also be represented in pseudocode as follows:</t>


<!-- tag::listcontinuationblock1[] -->


<ol type= "1">
<li>Take out the Holy Pin</li>
<li>
<t>The Count</t>
<figure>
<sourcecode><![CDATA[
```

```
integer count;
for count := 1 step 1 until 3 do
say(count)
comment Five is Right Out
]]></sourcecode>
</figure>
</li>
<li>Lob the Holy Hand Grenade of Antioch towards the Foe</li>
<li>Foe snuffs it</li>
</ol>


<!-- end::listcontinuationblock1[] -->

</section></section>
<section anchor= "_dramatis_personae" numbered= "true"><name>Dramatis
Personae</name><t>The following human (more-or-less) protagonists were
involved
in the two incidents recounted as lore of the Knights of the
Round Table:</t>


<!-- tag::table1[] -->


<table>
<thead>
<tr>
<th align= "left">French Castle</th>
<th align= "left">Cave of Caerbannog</th>
</tr>
</thead>
<tbody>
<tr>
<td colspan= "2" align= "left">King Arthur</td>
</tr>
<tr>
<td colspan= "2" align= "left">Patsy</td>
</tr>
<tr>
<td colspan= "2" align= "left">Sir Bedevere the Wise</td>
</tr>
<tr>
<td colspan= "2" align= "left">Sir Galahad the Pure</td>
</tr>
<tr>
<td colspan= "2" align= "left">Sir Lancelot the Brave</td>
</tr>
```

```
<tr>
<td colspan= "2" align= "left">Sir Robin the
Not-quite-so-brave-as-Sir-Lancelot</td>
</tr>
<tr>
<td align= "left">French Guard with Outrageous Accent</td>
<td align= "left">Tim the Enchanter</td>
</tr>
<tr>
<td align= "left">Other French Guards</td>
<td align= "left">Brother Maynard</td>
</tr>
<tr>
<td align= "left"/>
<td align= "left">The Lector</td>
</tr>
<tr>
<td rowspan= "3" align= "center">not yet recruited</td>
<td align= "right">Sir Bors</td>
</tr>
<tr>
<td align= "right">Sir Gawain</td>
</tr>
<tr>
<td align= "right">Sir Ector</td>
</tr>
</tbody>
<tfoot>
<tr>
<td align= "left">Retinue of sundry knights</td>
<td align= "left">Retinue of sundry more knights than at the
French Castle</td>
</tr>
</tfoot>
</table>


<!-- end::table1[] -->


<section anchor= "_past_the_killer_rabbit" numbered= "true"><name>Past
the Killer Rabbit</name><t>Once the Killer Rabbit of Caerbannog (<xref
target= "killer-bunny"/>) had been
dispatched, the Knights of the Round Table uncovered the last
words of Joseph of Arimathea, inscribed on the Cave of Caerbannog
in Aramaic.  While the precise Aramaic wording has not survived,
we trust the following Hebrew subtitles will serve as an
acceptable substitute:</t>
```

```
<!-- tag::hebrew1[] -->


<blockquote><t>&#1499;&#1488;&#1503; &#1488;&#1493;&#1500;&#1497;
&#1497;&#1502;&#1510;&#1488;&#1493;
&#1492;&#1502;&#1497;&#1500;&#1497;&#1501;
&#1492;&#1488;&#1495;&#1512;&#1493;&#1504;&#1493;&#1514; &#1513;&#1500;
&#1497;&#1493;&#1505;&#1507;
.&#1502;&#1488;&#1512;&#1502;&#1514;&#1497;&#1492;
&#1502;&#1497; &#1488;&#1513;&#1512; &#1497;&#1492;&#1497;&#1492;
&#1488;&#1502;&#1497;&#1509; &#1493;&#1489;&#1506;&#1500;
&#1504;&#1508;&#1513; &#1496;&#1492;&#1493;&#1512;&#1492;
&#1497;&#1493;&#1499;&#1500; &#1500;&#1502;&#1510;&#1493;&#1488;
&#1488;&#1514; &#1492;&#1490;&#1489;&#1497;&#1506;
&#1492;&#1511;&#1491;&#1493;&#1513; &#1489;&#1496;&#1497;&#1512;&#1514;
.&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1488;&#1492;</t>
<t>"Here may be found the last words of Joseph of Arimathea.
He who is valiant and pure of spirit may find the Holy Grail
in the castle of &#8212; Aaaargh."</t></blockquote>


<!-- end::hebrew1[] -->

</section></section>
<section anchor= "_iana_considerations" numbered= "true">
<name>IANA Considerations</name>
<t>IANA might consider a registry to track the mythical, especially
ravaging beasts, such as the Killer Rabbit, who haunt the Internet.</t>
</section>
<section anchor= "_security_considerations" numbered=
"true"><name>Security Considerations</name><t>Do not let the Killer
Rabbit out under any circumstance.</t>
<t>I repeat. Do not let the Killer Rabbit (<xref target=
"killer-bunny"/>) out.</t>


<!-- tag::bibliography1[] -->

</section>
</middle><back>
<references anchor= "_normative_references">
<name>Normative References</name>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2119.xml"
parse= "text"/>
</references>
<references anchor= "_informative_references">
```

```
<name>Informative References</name>
<reference anchor= "grail_film">
<front>
<title>Monty Python and the Holy Grail</title>
<author initials= "G." surname= "Chapman"/>
<author initials= "J." surname= "Cleese"/>
<author initials= "E." surname= "Idle"/>
<author initials= "T." surname= "Gilliam"/>
<author initials= "T." surname= "Jones"/>
<author initials= "M." surname= "Palin"/>
<date year= "1975"/>
</front>
</reference>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2635.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7990.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8140.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8174.xml"
parse= "text"/>
</references>
</back>
</rfc>
<CODE ENDS>
```

A.3. Example 3: "An API For Calendar-Based Fortune Heuristics Services" in AsciiRFC

This example is available in the following formats:

- AsciiRFC [git-divination-cfapi]

- Internet-Draft [I-D.divination-cfapi]

- Text, RFC XML, PDF and HTML on the IETF Datatracker [datatracker-divination-cfapi]

A.3.1. In AsciiRFC

```
<CODE BEGINS>
= An API For Calendar-Based Fortune Heuristics Services
Gabriel Destiny; Charise Luck
:doctype: internet-draft
:abbrev: Calendar Fortune Heuristics API
:name: draft-divination-cfapi-00
:status: informational
:ipr: trust200902
:area: Internet
:submission-type: independent
:intended-series: informational
:revdate: 2018-03-23T00:00:00Z
:lastname: Destiny
:fullname: Gabriel Destiny
:forename_initials: G.
:organization: Divination Inc.
:email: gabriel.destiny@ribose.com
:street: 9288 N Divine Street
:city: Dunn
:code: 28334
:region: NC
:country: United States of America
:lastname_2: Luck
:fullname_2: Charise Luck
:forename_initials_2: C.
:organization_2: Divination Inc.
:email_2: charise.luck@ribose.com
:street_2: 9288 N Divine Street
:city_2: Dunn
:code_2: 28334
:region_2: NC
:country_2: United States of America

[.comment]
tag::sample[]
// tag::sample[]

[abstract]

This document describes a JSON HTTP API for online services that
provide calendar-based fortune heuristics.

== Introduction

Fortune-telling, the practice of predicting information about a
person's life, is an activity practiced throughout history.
```

While there are myriad forms of fortune telling methodologies, this document applies to a particular form of service that provides fortune heuristics, commonly known as "luck", for a particular subject based on a calendar-based input.

Since HTTP <<RFC7230>> status codes are insufficient to convey information about fortune heuristics, this specification defines a simple JSON <<RFC8259>> document format for this purpose. The response can be used by HTTP APIs to deliver results to non-human clients or to an end-user.

== Conventions Used in This Document

The key words "*MUST*", "*MUST NOT*", "*REQUIRED*", "*SHALL*", "*SHALL NOT*", "*SHOULD*", "*SHOULD NOT*", "*RECOMMENDED*", "*NOT RECOMMENDED*", "*MAY*", and "*OPTIONAL*" in this document are to be interpreted as described in BCP 14 <<RFC2119>> <<RFC8174>> when, and only when, they appear in all capitals, as shown here.

The following definitions apply in this document:

Well-known URI:: This specification makes use of the "well-known URI" feature of HTTP servers <<RFC5785>> to provide a bootstrapping URI for the client usage of fortune heuristics services.

Root of Fortune:: The service discovery endpoint that provides a URI list of available fortune heuristic endpoints available, in accordance with <<service-discovery>>.

== Fortune Heuristics Service Well-Known URI

A well-known URI called "fortune" is registered by this specification for fortune heuristics services (see <<iana>>).

Services complying with this document *SHOULD* have its well-known URI pointing (directly or through redirection) to the Root of Fortune.

The Root of Fortune can be used by the client for service discovery, namely, the available calendar-based fortune heuristics services available on the server, as specified in <<service-discovery>>.

=== Well-Known URI Redirection

Servers *MUST* redirect HTTP requests for that resource to the actual "context path" using one of the available mechanisms provided by HTTP <<RFC7230>> (e.g., using a 301, 303, or 307 response).

Clients *MUST* handle HTTP redirects on the well-known URI.

=== Well-Known URI Cache Behavior

Servers *SHOULD* set an appropriate Cache-Control header value (as according to <<RFC7234,5.2 of>>) in the redirect response to set caching behavior as required by the type of response generated.

== New HTTP Methods: SEEK and DIVINE

This specification defines two new HTTP methods: "SEEK" and "DIVINE" methods for HTTP <<RFC7230>>.

While HTTP GET requests are treated equivalently as the "SEEK" and "DIVINE" requests, its usage is discouraged and therefore *SHOULD NOT* be used.

Usage of these methods are defined in the sections below.

== Defined Data Types: Date-Time Formats

This specification defines a number of date-time formats as according to the conventions of <<RFC3339>> for the unambiguous communication between client and server.

The types defined are as follows.

`DATETIME`::
As described in <<RFC3339,5.6 of>>, with the addition that reduced accuracy representations described in <<ISO.8601-1.2018>> are supported. Reduced accuracy date and times are accepted where a date or time component (2-digits long) is replaced by "--".
+
For example, the date time "2018-04---T01:02:00Z" represents the UTC time of 1:02am, on an unknown day within April of the year 2018.

`DATE`::
As described in "DATETIME", but the "time" component will not be taken into account in the algorithm.

[#service-discovery]
== Fortune Heuristics Service Discovery

[#root-of-fortune]

=== Root of Fortune Path URI ("/")

The Root of Fortune URI, defined as "/" in this document, is used for service discovery on types of calendar-based fortune heuristics available.

An empty SEEK request with the "application/json" request type *MUST* be sent to this endpoint to retrieve the available endpoints. All other HTTP methods *MUST NOT* be supported at this URI.

An example of such a response is as follows:

```
[source,json]
----
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"diviners" : [
"/astrology",
"/bazi",
]
}
----
```

A service discovery object *MUST* have the following members:

`diviners`::
(JSON array)
An array that contains endpoints that conform to this specification. All endpoints listed here are relative to the Root of Fortune path. For example, the path "/astrology" listed in the example has an endpoint path of "[root-of-fortune]/astrology", where "[root-of-fortune]" indicates the real path of the Root of Fortune.

```
// end::sample[]
[.comment]
end::sample[]
```

[#service-endpoint]
== Fortune Heuristics Service Endpoint

An endpoint offering fortune heuristics services *MUST* adhere to specifications in this section.

A service *MAY* implement multiple divination services based on different divination methods, such as the digital oracle shown in <<digital-oracle>>.

[[digital-oracle]]
.Dimensional Eye, a digital oracle that communicates through one button
====
[alt=An incarnation of the Dimensional Eye, in ASCII]
....

```
                                  __
                              __===^-\
                         __===       -\
                     __===-           -\
                __===-                 -\
           ___===-                      -\
        ===-                ---__        -\
        \\\              |||^^\ \__       -\
         \\\             |||       \__      -\
          \\\            |||    ______\_     -\
           \\\           |||  _/-******\\__   -\
            \\\          ||  /-****_****-\ \_  -\
             \\\         || |-***/   \***-\  \_ -\
              \\\        || |-***\___/***-|    \ -\
               \\\       ||  \-*********-/  __--/ -\
                \\\      ||    \-******/__--       -\
                 \\\     ||        __--             -\
                  \\\    ||    __--                  -\
                   \\\   ||__--                       -\
                    \\\                                -\
                     \\\                                -\
                      \\\                                -\
                       \\\                                -\
                        \\\                 __             -\
                         \\\              //##\\            -\
                          \\\             \\##//             -\
                           \\\              ^^          __--==^
                            \\\                   __--===
                             \\\             __--===
                              \\\      __--===
                               \\\ __--==
                                \\=
```

....
====

[#endpoint-specification-request]
=== Service Specification Request

To retrieve capabilities and parameters of an endpoint complying with this specification, a service specification JSON object is returned.

An empty SEEK request with the "application/json" request type *MUST* be sent to this endpoint to retrieve the service specification that describes parameters accepted by this endpoint.

Two examples of such a response are given below.

[source,json]
```
----
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"description": "Gaze into your upcoming luck!",
"details": "https://divine.example.com/manual/astrology-api",
"parameters": {
"birthday": {
"type": "DATE",
"description": "Your birth date in UTC"
},
"targetDateBegin": {
"type": "DATE",
"description": "Start of the target date range to report on"
},
"targetDateEnd": {
"type": "DATE",
"description": "End of the target date range to report on"
},
"interval": {
"values": {
"D": "Daily",
"M": "Monthly",
"Y": "Yearly"
},
"description": "Available intervals to report on."
}
}
}
----
```

[source,json]
```
----
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en
```

```
{
"description": "Matches and mis-matches according to the "
"Yin Yang and Five Elements techniques",
"details": "https://divine.example.com/manual/bazi-api",
"parameters": {
"birthday": {
"type": "DATETIME",
"description": "Your birth date and time in UTC"
},
"targetDateBegin": {
"type": "DATETIME",
"description": "Start of the target date/time range to report on"
},
"targetDateEnd": {
"type": "DATETIME",
"description": "End of the target date/time range to report on"
},
"interval": {
"values": {
"H": "Hourly",
"D": "Daily",
"M": "Monthly",
"Y": "Yearly"
},
"description": "Available intervals to report on."
}
}
}
----

[#service-endpoint-specification]
=== Service Specification Object

A service specification object *MUST* contain the following members.

`description`::
(string) A short, human-readable summary of the fortune heuristic
service at this endpoint. This *SHOULD* be a stable reference.

`details`::
(URI, optional) A URI reference that provides further details for
human consumption, such as API documentation that includes details of
parameters accepted or response states.

`parameters`::
(object, mandatory) An object that specifies what parameters
are accepted by this endpoint. Each parameter key within this object
specifies an accepted parameter name, and its value is a parameter
```

specification object, which is described below.

A parameter specification object *SHOULD* contain the following members:

`type`::
(string, optional) The value type accepted by this parameter. Value types are described in this document. This member is mutually exclusive with `values`.

`description`::
(string, mandatory) The purpose of this parameter.

`values`::
(object, optional) The accepted values of this parameter, unlisted values *SHOULD* not be accepted by the parameter. Each key within this object specifies an accepted value, and its value provides a description of the purpose of the value.

[#endpoint-report]
== Fortune Heuristics Report Request and Response

[#endpoint-report-request]
=== Fortune Heuristics Report Request

A request using the HTTP "DIVINE" method and the "application/json" type *MUST* be sent to the fortune heuristic endpoint to retrieve results for a fortune heuristic query.

The request made *MUST* conform to the specifications of the endpoint, as retrieved via the "SEEK" method described in <<endpoint-specification-request>>.

An example of a request is provided below.

[source]
```
URI: /divination/astrology
Method: DIVINE
Content-Type: application/json
Content-Language: en

{
"birthday": "1976-02-11T00:00:00Z",
"targetDateBegin": "2018-01-01T00:00:00Z",
"targetDateEnd": "2019-01-01T00:00:00Z",
"interval": "M"
```

```
}
----


[#endpoint-report-response]
=== Fortune Heuristics Report Response

A fortune heuristic query using the "DIVINE" method triggers a
response that contains a fortune heuristics report.

A successful response returns a JSON object that *MUST* conform
to the object structure described in this section.

A report object *SHOULD* contain the following members:

`type`::
(URI, mandatory) A URI that defines the type of the report located
at the `report` key of this object.

`report`::
(object, mandatory) An object that contains two keys, `intervals`
and `events`. The `intervals` object contains an array of interval
objects that matches the demanded intervals in the request within
the target date range.
The `events` object contains an array of significant event objects
within the target date range.

An example of a response is provided below.

[source]
----
URI: /divination/astrology
Method: DIVINE
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"type": "https://association-of.astrology/reports/monthly",
"report": {
"intervals": [
{
"dateStart": "2018-01-01T00:00:00Z",
"dateEnd": "2018-02-01T00:00:00Z",
"categories": [
{
"category":
"https://divine.example.com/astrology/categories/health"
```

```
"value": 80,
"description": "Charge ahead with excellent health."
},
{
"category":
"https://divine.example.com/astrology/categories/love"
"value": 70,
"description":
"Give a certain person or situation another try!"
},
{
"category":
"https://divine.example.com/astrology/categories/finance"
"value": 5,
"description": "You've just realized that you don't have
any cash on hand."
}
]
},
{
"dateStart": "2018-02-01T00:00:00Z",
"dateEnd": "2018-03-01T00:00:00Z",
"..."
},
"..."
],
"events": [
{
"dateStart": "2018-01-15T03:20:00",
"dateEnd": "2018-01-16T20:22:15",
"description": "The planet of growth and good luck, Jupiter
will make a harmonious connection with power planet Pluto,
helping you connect with influential people",
"recommendation": "Engage in networking during this time."
},
{
"dateStart": "2018-03-22T00:12:40",
"dateEnd": "2018-03-28T02:45:03",
"description": "Communication planet Mercury enters your sign,
which will find you in a busier and chattier mood.",
"recommendation":
"Take charge of work with your newfound energy."
}
"..."
]
}
}
----
```

Fortune heuristic reports are created by a divination output that *MAY* requires quantitative interpretation. A sample representation of interpreting a graphical divination output is provided in <<divination-message>>.

[[divination-message]]
.Forty-nine yarrow sticks reveals a mystical message on fortune
====
[alt=A mystical pattern in ASCII]
....

```
                                    0000000000000000000000000
        0000000000000000000000001 G 1000000000000000000000000
        0000000000000000000000011 R 1100000000000000000000000
        0000000000000000000000111 A 1110000000000000000000000
        0000000000000000000001111 C 1111000000000000000000000
        0000000000000000000011111 E 1111100000000000000000000
        0000000000000000000111111 , 1111110000000000000000000
        0000000000000000001111111   1111111000000000000000000
        0000000011111111111111111 M 1111111111111111100000000
        0000000111111111111111111 E 1111111111111111110000000
        0000001111111111111111111 R 1111111111111111111000000
        0000011111111111111111111 C 1111111111111111111100000
        0000111111111111111111111 Y 1111111111111111111110000
        0001111111111111111111111 , 1111111111111111111111000
        0011111111111111111111111   1111111111111111111111100
        0111111111111111111111111 A 1111111111111111111111110
        0000000000000000011111111 N 1111111100000000000000000
        0000000000000000111111111 D 1111111110000000000000000
        0000000000000001111111111   1111111111000000000000000
        0000000000000011111111111 P 1111111111100000000000000
        0000000000000111111111111 E 1111111111110000000000000
        0000000000001111111111111 A 1111111111111000000000000
        0000000000011111111111111 C 1111111111111100000000000
        0000000000111111111111111 E 1111111111111110000000000
        0000000001111111111111111 . 1111111111111111000000000
```

....
====

[#endpoint-report-interval-obj]
=== Report Interval Object

The `intervals` value of a report object contains a number of report intervals -- each representing a non-overlapping period of the selected interval length. When all of these intervals are put together, the combined period *MUST* fully cover the requested report target period.

An example interval object is shown below.

```
[source,json]
----
{
"dateStart": "2018-01-01T00:00:00Z",
"dateEnd": "2018-02-01T00:00:00Z",
"categories": [
{
"category":
"https://divine.example.com/astrology/categories/health"
"value": 80,
"description": "Charge ahead with your excellent health."
},
{
"category":
"https://divine.example.com/astrology/categories/love"
"value": 70,
"description": "Give a certain person or situation another try!"
},
{
"category":
"https://divine.example.com/astrology/categories/finance"
"value": 5,
"description": "You've just realized that you don't have
any cash on hand."
}
]
}
----
```

An interval object *MUST* contain the following members:

`dateStart`::
(datetime, mandatory) This value specifies the start of the period which this interval object applies to.

`dateEnd`::
(datetime, mandatory) This value specifies the end of the period which this interval object applies to.

In the given example, the `categories` key is an implementation specific object that details heuristic results returned by the selected algorithm. This *MAY* differ in different algorithms.

[#endpoint-report-event-obj]
=== Report Events Object

The `events` value of a report object contains a number of event objects. Each event object represents an event relevant to the calculation of fortune heuristics during a target report period. These events *MAY* be of variable time lengths, and *MAY* be overlapping amongst each other.

The following example demonstrates two event objects the service determines relevant to a user's query.

```
[source,json]
----
{
"dateStart": "2018-01-15T03:20:00",
"dateEnd": "2018-01-16T20:22:15",
"description": "The planet of growth and good luck, Jupiter will
make a harmonious connection with power planet Pluto, helping you
connect with influential people",
"recommendation": "Engage in networking during this time."
},
{
"dateStart": "2018-03-22T00:12:40",
"dateEnd": "2018-03-28T02:45:03",
"description": "Communication planet Mercury enters your sign,
which will find you in a busier and chattier mood.",
"recommendation": "Take charge of work with your newfound energy."
}
----
```

Similar to an interval object, an event object *MUST* contain the following members:

`dateStart`::
(datetime, mandatory) This value specifies the start of the period described by the event.

`dateEnd`::
(datetime, mandatory) This value specifies the end of the period described by the event.

In the given example, the keys `description` and `recommendation` are implementation-specific details. This *MAY* differ in different algorithms.

[#endpoint-report-errors]
=== Report Generation Errors

This specification makes use of normal HTTP error codes with the

following extensions.

Errors *MUST* be returned using the Problem JSON Structure as accordance with <<RFC7807>>.

422 Unprocessable Entity::
For example, a malformed date-time parameter, or an illogical input, such as when the subject's birthday occurs after the report target date period.

473 Beyond Existing Capability::
The service determines that the outcome is too difficult to predict. For example, in the case where the calculation is too complex to complete in a certain time period. The service *SHOULD* issue this response code to indicate that the client should not try the same request again.

474 Outcome Impossible::
The service determines that the outcome is impossible. For example, when the algorithm determines that the subject will have deceased before the start of the requested target period.

[#security]
== Security Considerations

* TLS <<RFC5246>> and authenticated HTTP requests should be used to protect the DIVINE request and responses due to the personal nature of information transmitted.

* A client *SHOULD* verify the identity of the server on every request to prevent impersonation or man-in-the-middle attacks, as data transmitted to and from the server is sensitive information, and at times critical information to the user.

* Synchronization of client and server time *MUST* be well-considered in the implementation of this specification. A mismatch of client and server time *MAY* lead to algorithm miscalculations that can cause mistaken choices of a user that depends on the reliability of this system.

[#iana]
== IANA Considerations

=== Well-Known URI Registrations

```
This document defines a well-known URI using the registration
procedure and template from <<RFC5785,5.1 of>>.

==== "fortune" Well-Known URI Registration

URI suffix::  fortune

Change controller::  IETF

Specification document(s)::  This document

Related information::  N/A.


[.comment]
tag::sample[]
// begin::sample[]

[bibliography]
== Normative References

++++

<reference anchor= 'RFC2119'
target= 'https://www.rfc-editor.org/info/rfc2119'>
<front>
<title>Key words for use in RFCs to Indicate Requirement
Levels</title>
<author initials= 'S.' surname= 'Bradner' fullname='S. Bradner'>
<organization />
</author>
<date year= '1997' month= 'March' />
<abstract><t>In many standards track documents several words are
used to signify the requirements in the specification.  These
words are often capitalized. This document defines these words
as they should be interpreted in IETF documents.  This
document specifies an Internet Best Current Practices for the
Internet Community, and requests discussion and suggestions
for improvements.</t></abstract>
</front>
<seriesInfo name= 'BCP' value= '14'/>
<seriesInfo name= 'RFC' value= '2119'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC2119'/>
</reference>

<reference anchor= 'RFC5785'
target= 'https://www.rfc-editor.org/info/rfc5785'>
<front>
```

```
<title>Defining Well-Known Uniform Resource Identifiers
(URIs)</title>
<author initials= 'M.' surname= 'Nottingham'
fullname='M. Nottingham'>
<organization />
</author>
<author initials= 'E.' surname= 'Hammer-Lahav'
fullname='E. Hammer-Lahav'>
<organization />
</author>
<date year= '2010' month= 'April' />
<abstract><t>This memo defines a path prefix for &quot;well-known
locations&quot;, &quot;/.well-known/&quot;, in selected Uniform
Resource Identifier (URI) schemes.
[STANDARDS-TRACK]</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '5785'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC5785'/>
</reference>

<reference anchor= 'RFC7230'
target= 'https://www.rfc-editor.org/info/rfc7230'>
<front>
<title>Hypertext Transfer Protocol (HTTP/1.1): Message Syntax and
Routing</title>
<author initials= 'R.' surname= 'Fielding' fullname='R. Fielding'
role='editor'>
<organization />
</author>
<author initials= 'J.' surname= 'Reschke' fullname='J. Reschke'
role='editor'>
<organization />
</author>
<date year= '2014' month= 'June' />
<abstract><t>The Hypertext Transfer Protocol (HTTP) is a stateless
application-level protocol for distributed, collaborative,
hypertext information systems.  This document provides an
overview of HTTP architecture and its associated terminology,
defines the &quot;http&quot; and &quot;https&quot; Uniform
Resource Identifier (URI) schemes, defines the HTTP/1.1
message syntax and parsing requirements, and describes related
security concerns for implementations.</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '7230'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC7230'/>
</reference>

<reference anchor= 'RFC7234'
```

```
target= 'https://www.rfc-editor.org/info/rfc7234'>
<front>
<title>Hypertext Transfer Protocol (HTTP/1.1): Caching</title>
<author initials= 'R.' surname= 'Fielding' fullname='R. Fielding'
role='editor'>
<organization />
</author>
<author initials= 'M.' surname= 'Nottingham' fullname='M.
Nottingham'
role='editor'>
<organization />
</author>
<author initials= 'J.' surname= 'Reschke' fullname='J. Reschke'
role='editor'>
<organization />
</author>
<date year= '2014' month= 'June' />
<abstract><t>The Hypertext Transfer Protocol (HTTP) is a stateless
\%application- level protocol for distributed, collaborative,
hypertext information systems.  This document defines HTTP
caches and the associated header fields that control cache
behavior or indicate cacheable response
messages.</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '7234'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC7234'/>
</reference>

<reference anchor= 'RFC7807'
target= 'https://www.rfc-editor.org/info/rfc7807'>
<front>
<title>Problem Details for HTTP APIs</title>
<author initials= 'M.' surname= 'Nottingham'
fullname='M. Nottingham'>
<organization />
</author>
<author initials= 'E.' surname= 'Wilde' fullname='E. Wilde'>
<organization />
</author>
<date year= '2016' month= 'March' />
<abstract><t>This document defines a &quot;problem detail&quot;
as a way to carry machine- readable details of errors in a
HTTP response to avoid the need to define new error response
formats for HTTP APIs.</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '7807'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC7807'/>
</reference>
```

```
<reference anchor= 'RFC8174'
target= 'https://www.rfc-editor.org/info/rfc8174'>
<front>
<title>Ambiguity of Uppercase vs Lowercase in RFC 2119 Key
Words</title>
<author initials= 'B.' surname= 'Leiba' fullname='B. Leiba'>
<organization />
</author>
<date year= '2017' month= 'May' />
<abstract><t>RFC 2119 specifies common key words that may be used
in protocol specifications.  This document aims to reduce
the ambiguity by clarifying that only UPPERCASE usage of the
key words have the defined special meanings.</t></abstract>
</front>
<seriesInfo name= 'BCP' value= '14'/>
<seriesInfo name= 'RFC' value= '8174'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC8174'/>
</reference>

<reference anchor= 'RFC8259'
target= 'https://www.rfc-editor.org/info/rfc8259'>
<front>
<title>The JavaScript Object Notation (JSON) Data Interchange
Format</title>
<author initials= 'T.' surname= 'Bray' fullname='T. Bray'
role='editor'>
<organization />
</author>
<date year= '2017' month= 'December' />
<abstract><t>JavaScript Object Notation (JSON) is a lightweight,
text-based, language-independent data interchange format.
It was derived from the ECMAScript Programming Language
Standard.  JSON defines a small set of formatting rules for
the portable representation of structured data.</t>
<t>This document removes inconsistencies with other
specifications of JSON, repairs specification errors, and
offers experience-based interoperability
guidance.</t>
</abstract>
</front>
<seriesInfo name= 'STD' value= '90'/>
<seriesInfo name= 'RFC' value= '8259'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC8259'/>
</reference>
++++

[bibliography]
== Informative References
```

```
++++

<reference anchor= 'ISO.8601-1.2018'
target= 'https://www.iso.org/en/standard/70907.html'>
<front>
<title>ISO/DIS 8601-1:2018, Data elements and interchange
formats -- Information interchange -- Representation of dates
and times -- Part 1: Basic rules</title>
<author>
<organization>ISO/IEC</organization>
<address>
<uri>http://www.iso.org</uri>
</address>
</author>
<date month= 'January' year= '2018'/>
<abstract><t></t></abstract>
</front>
</reference>


<reference anchor= 'RFC3339'
target= 'https://www.rfc-editor.org/info/rfc3339'>
<front>
<title>Date and Time on the Internet: Timestamps</title>
<author initials= 'G.' surname= 'Klyne' fullname='G. Klyne'>
<organization />
</author>
<author initials= 'C.' surname= 'Newman' fullname='C. Newman'>
<organization />
</author>
<date year= '2002' month= 'July' />
</front>
<seriesInfo name= 'RFC' value= '3339'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC3339'/>
</reference>

<reference anchor= 'RFC5246'
target= 'https://www.rfc-editor.org/info/rfc5246'>
<front>
<title>The Transport Layer Security (TLS) Protocol
Version 1.2</title>
<author initials= 'T.' surname= 'Dierks' fullname='T. Dierks'>
<organization />
</author>
<author initials= 'E.' surname= 'Rescorla' fullname='E. Rescorla'>
<organization />
</author>
<date year= '2008' month= 'August' />
```

```
<abstract><t>This document specifies Version 1.2 of the Transport
Layer Security (TLS) protocol.  The TLS protocol provides
communications security over the Internet.  The protocol
allows client/server applications to communicate in a way
that is designed to prevent eavesdropping, tampering, or
message forgery.  [STANDARDS-TRACK]</t></abstract>
</front>
<seriesInfo name= 'RFC' value= '5246'/>
<seriesInfo name= 'DOI' value= '10.17487/RFC5246'/>
</reference>
++++



[appendix]
== Acknowledgements

The authors thank the following individuals for their valuable
feedback on this specification, and commend them for making fortune
heuristics more accessible for the benefit of mankind.

// end::sample[]
[.comment]
end::sample[]
<CODE ENDS>
```

A.4. Rendered as RFC XML v3

```
<CODE BEGINS>
<?xml version= "1.0" encoding= "US-ASCII"?>
<?xml-stylesheet type= "text/xsl" href= "rfc2629.xslt"?>
<!DOCTYPE rfc SYSTEM "rfc2629.dtd">
<?rfc strict= "yes"?>
<?rfc compact= "yes"?>
<?rfc subcompact= "no"?>
<?rfc toc= "yes"?>
<?rfc tocdepth= "4"?>
<?rfc symrefs= "yes"?>
<?rfc sortrefs= "yes"?>
<rfc xmlns:xi= "http://www.w3.org/2001/XInclude" ipr= "trust200902"
submissionType="independent" prepTime="2018-04-18T03:35:38Z"
version="3">
<front>
<title abbrev= "Calendar Fortune Heuristics API">An API For
Calendar-Based Fortune Heuristics Services</title>
<seriesInfo name= "Internet-Draft" status= "informational"
stream="independent" value="draft-divination-cfapi-00"/>
<seriesInfo name= "" status="informational"
```

```
value="draft-divination-cfapi-00"/>
<author fullname= "Gabriel Destiny" surname= "Destiny" initials="G.">
<organization>Divination Inc.</organization>
<address>
<postal>
<street>9288 N Divine Street</street>
<city>Dunn</city>
<region>NC</region>
<code>28334</code>
<country>United States of America</country>
</postal>
<email>gabriel.destiny@ribose.com</email>
</address>
</author>
<author fullname= "Charise Luck" surname= "Luck" initials="C.">
<organization>Divination Inc.</organization>
<address>
<postal>
<street>9288 N Divine Street</street>
<city>Dunn</city>
<region>NC</region>
<code>28334</code>
<country>United States of America</country>
</postal>
<email>charise.luck@ribose.com</email>
</address>
</author>
<date day= "23" month= "March" year="2018"/>
<area>Internet</area>

<abstract>

<!-- tag::sample[] -->


<t>This document describes a JSON HTTP API for online services that
provide calendar-based fortune heuristics.</t></abstract>
</front><middle>
<section anchor= "_introduction" numbered=
"false"><name>Introduction</name><t>Fortune-telling, the practice of
predicting information about a
person&#8217;s life, is an activity practiced throughout history.</t>
<t>While there are myriad forms of fortune telling methodologies, this
document applies to a particular form of service that provides fortune
heuristics, commonly known as "luck", for a particular subject based
on a calendar-based input.</t>
<t>Since HTTP <xref target= "RFC7230"/> status codes are insufficient to
convey
```

```
information about fortune heuristics, this specification defines a
simple JSON <xref target= "RFC8259"/> document format for this purpose.
The
response can be used by HTTP APIs to deliver results to non-human
clients or to an end-user.</t></section>
<section anchor= "_conventions_used_in_this_document" numbered=
"false"><name>Conventions Used in This Document</name><t>The key words
"<bcp14>MUST</bcp14>", "<bcp14>MUST NOT</bcp14>",
"<bcp14>REQUIRED</bcp14>", "<bcp14>SHALL</bcp14>",
"<bcp14>SHALL NOT</bcp14>", "<bcp14>SHOULD</bcp14>", "<bcp14>SHOULD
NOT</bcp14>", "<bcp14>RECOMMENDED</bcp14>",
"<strong>NOT RECOMMENDED</strong>", "<bcp14>MAY</bcp14>", and
"<bcp14>OPTIONAL</bcp14>" in this document
are to be interpreted as described in BCP 14 <xref target= "RFC2119"/>
<xref target="RFC8174"/>
when, and only when, they appear in all capitals, as shown here.</t>
<t>The following definitions apply in this document:</t>
<dl>
<dt>Well-known URI</dt>
<dd>This specification makes use of the "well-known URI"
feature of HTTP servers <xref target= "RFC5785"/> to provide a
bootstrapping URI for
the client usage of fortune heuristics services.</dd>
<dt>Root of Fortune</dt>
<dd>The service discovery endpoint that provides a URI
list of available fortune heuristic endpoints available, in accordance
with <xref target= "service-discovery"/>.</dd>
</dl></section>
<section anchor= "_fortune_heuristics_service_well_known_uri" numbered=
"false"><name>Fortune Heuristics Service Well-Known URI</name><t>A
well-known URI called "fortune" is registered by this specification
for fortune heuristics services (see <xref target= "iana"/>).</t>
<t>Services complying with this document <bcp14>SHOULD</bcp14> have its
well-known
URI pointing (directly or through redirection) to the Root of
Fortune.</t>
<t>The Root of Fortune can be used by the client for service discovery,
namely, the available calendar-based fortune heuristics services
available on the server, as specified in <xref target=
"service-discovery"/>.</t>
<section anchor= "_well_known_uri_redirection" numbered=
"false"><name>Well-Known URI Redirection</name><t>Servers
<bcp14>MUST</bcp14> redirect HTTP requests for that resource to the
actual "context path" using one of the available mechanisms provided
by HTTP <xref target= "RFC7230"/> (e.g., using a 301, 303, or 307
response).</t>
<t>Clients <bcp14>MUST</bcp14> handle HTTP redirects on the well-known
URI.</t></section>
```

```
<section anchor= "_well_known_uri_cache_behavior" numbered= "false">
<name>Well-Known URI Cache Behavior</name>
<t>Servers <bcp14>SHOULD</bcp14> set an appropriate Cache-Control
header value (as
according to <relref section= "5.2" displayFormat= "of"
target="RFC7234"/>) in the redirect response to set
caching behavior as required by the type of response generated.</t>
</section></section>
<section anchor= "_new_http_methods_seek_and_divine" numbered=
"false"><name>New HTTP Methods: SEEK and DIVINE</name><t>This
specification defines two new HTTP methods: "SEEK" and "DIVINE"
methods for HTTP <xref target= "RFC7230"/>.</t>
<t>While HTTP GET requests are treated equivalently as the "SEEK" and
"DIVINE" requests, its usage is discouraged and therefore <bcp14>SHOULD
NOT</bcp14>
be used.</t>
<t>Usage of these methods are defined in the sections
below.</t></section>
<section anchor= "_defined_data_types_date_time_formats" numbered=
"false"><name>Defined Data Types: Date-Time Formats</name><t>This
specification defines a number of date-time formats as according
to the conventions of <xref target= "RFC3339"/> for the unambiguous
communication
between client and server.</t>
<t>The types defined are as follows.</t>
<dl>
<dt>
<tt>DATETIME</tt>
</dt>
<dd>
<t>As described in <relref section= "5.6" displayFormat= "of"
target="RFC3339"/>, with the addition that reduced
accuracy representations described in <xref target= "ISO.8601-1.2018"/>
are
supported.  Reduced accuracy date and times are accepted where a
date or time component (2-digits long) is replaced by "--".</t>
<t>For example, the date time "2018-04---T01:02:00Z" represents the
UTC
time of 1:02am, on an unknown day within April of the year 2018.</t>
</dd>
<dt>
<tt>DATE</tt>
</dt>
<dd>As described in "DATETIME", but the "time" component will not be
taken into account in the algorithm.</dd>
</dl></section>
<section anchor= "service-discovery" numbered= "false">
<name>Fortune Heuristics Service Discovery</name>
```

```
<section anchor= "root-of-fortune" numbered= "false"><name>Root of
Fortune Path URI ("/")</name><t>The Root of Fortune URI, defined as "/"
in this document, is used for
service discovery on types of calendar-based fortune heuristics
available.</t>
<t>An empty SEEK request with the "application/json" request type
<bcp14>MUST</bcp14> be sent to this endpoint to retrieve the available
endpoints.
All other HTTP methods <bcp14>MUST NOT</bcp14> be supported at this
URI.</t>
<t>An example of such a response is as follows:</t>
<figure>
<sourcecode type= "json"><![CDATA[
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"diviners" : [
"/astrology",
"/bazi",
]
}
]]></sourcecode>
</figure>
<t>A service discovery object <bcp14>MUST</bcp14> have the following
members:</t>
<dl>
<dt>
<tt>diviners</tt>
</dt>
<dd>(JSON array)
An array that contains endpoints that conform to this specification.
All endpoints listed here are relative to the Root of Fortune path.
For example, the path "/astrology" listed in the example has an
endpoint path of "[root-of-fortune]/astrology", where
"[root-of-fortune]" indicates the real path of the Root of Fortune.</dd>
</dl>


<!-- end::sample[] -->

</section>
</section>
<section anchor= "service-endpoint" numbered= "false"><name>Fortune
Heuristics Service Endpoint</name><t>An endpoint offering fortune
heuristics services <bcp14>MUST</bcp14> adhere to
specifications in this section.</t>
```

```
<t>A service <bcp14>MAY</bcp14> implement multiple divination services
based on
different divination methods, such as the digital oracle shown
in <xref target= "digital-oracle"/>.</t>
<figure anchor= "digital-oracle">
<name>Dimensional Eye, a digital oracle that communicates through one
button</name>
<artwork type= "ascii-art" alt= "An incarnation of the Dimensional
Eye"><![CDATA[
                                 __
                             __===^-\
                        __===       -\
                    __===-           -\
                __===-                 -\
           ___===-                      -\
       ===-                ---__          -\
       \\\              |||^^\ \__        -\
        \\\             |||       \__      -\
         \\\            |||    ______\_     -\
          \\\           |||  _/-******\\__   -\
           \\\          ||  /-****_****-\ \_  -\
            \\\         || |-***/   \***-\  \_ -\
             \\\        || |-***\___/***-|    \ -\
              \\\       ||  \-*********-/  __--/ -\
               \\\      ||    \-******/__--       -\
                \\\     ||        __--              -\
                 \\\    ||    __--                  -\
                  \\\   ||__--                       -\
                   \\\                                -\
                    \\\                                -\
                     \\\                                -\
                      \\\                                -\
                       \\\                 __             -\
                        \\\              //##\\            -\
                         \\\             \\##//             -\
                          \\\              ^^          __--==^
                           \\\                    __--===
                            \\\             __--===
                             \\\      __--===
                              \\\ __--==
                               \\=

]]></artwork>
</figure>
<section anchor= "endpoint-specification-request" numbered=
"false"><name>Service Specification Request</name><t>To retrieve
capabilities and parameters of an endpoint complying with
this specification, a service specification JSON object is returned.</t>
```

```
<t>An empty SEEK request with the "application/json" request type
<bcp14>MUST</bcp14> be sent to this endpoint to retrieve the service
specification that describes parameters accepted by this endpoint.</t>
<t>Two examples of such a response are given below.</t>
<figure>
<sourcecode type= "json"><![CDATA[
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"description": "Gaze into your upcoming luck!",
"details": "https://divine.example.com/manual/astrology-api",
"parameters": {
"birthday": {
"type": "DATE",
"description": "Your birth date in UTC"
},
"targetDateBegin": {
"type": "DATE",
"description": "Start of the target date range to report on"
},
"targetDateEnd": {
"type": "DATE",
"description": "End of the target date range to report on"
},
"interval": {
"values": {
"D": "Daily",
"M": "Monthly",
"Y": "Yearly"
},
"description": "Available intervals to report on."
}
}
}
]]></sourcecode>
</figure>
<figure>
<sourcecode type= "json"><![CDATA[
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"description": "Matches and mis-matches according to the "
"Yin Yang and Five Elements techniques",
"details": "https://divine.example.com/manual/bazi-api",
```

```
"parameters": {
"birthday": {
"type": "DATETIME",
"description": "Your birth date and time in UTC"
},
"targetDateBegin": {
"type": "DATETIME",
"description": "Start of the target date/time range to report on"
},
"targetDateEnd": {
"type": "DATETIME",
"description": "End of the target date/time range to report on"
},
"interval": {
"values": {
"H": "Hourly",
"D": "Daily",
"M": "Monthly",
"Y": "Yearly"
},
"description": "Available intervals to report on."
}
}
}
]]></sourcecode>
</figure></section>
<section anchor= "service-endpoint-specification" numbered=
"false"><name>Service Specification Object</name><t>A service
specification object <bcp14>MUST</bcp14> contain the following
members.</t>
<dl>
<dt>
<tt>description</tt>
</dt>
<dd>(string) A short, human-readable summary of the fortune heuristic
service at this endpoint. This <bcp14>SHOULD</bcp14> be a stable
reference.</dd>
<dt>
<tt>details</tt>
</dt>
<dd>(URI, optional) A URI reference that provides further details for
human consumption, such as API documentation that includes details of
parameters accepted or response states.</dd>
<dt>
<tt>parameters</tt>
</dt>
<dd>(object, mandatory) An object that specifies what parameters
are accepted by this endpoint. Each parameter key within this object
```

```
specifies an accepted parameter name, and its value is a parameter
specification object, which is described below.</dd>
</dl>
<t>A parameter specification object <bcp14>SHOULD</bcp14> contain the
following
members:</t>
<dl>
<dt>
<tt>type</tt>
</dt>
<dd>(string, optional) The value type accepted by this parameter.
Value
types are described in this document. This member is mutually
exclusive with <tt>values</tt>.</dd>
<dt>
<tt>description</tt>
</dt>
<dd>(string, mandatory) The purpose of this parameter.</dd>
<dt>
<tt>values</tt>
</dt>
<dd>(object, optional) The accepted values of this parameter, unlisted
values <bcp14>SHOULD</bcp14> not be accepted by the parameter. Each key
within
this object specifies an accepted value, and its value provides a
description of the purpose of the value.</dd>
</dl></section></section>
<section anchor= "endpoint-report" numbered= "false"><name>Fortune
Heuristics Report Request and Response</name><section anchor=
"endpoint-report-request" numbered="false"><name>Fortune Heuristics
Report Request</name><t>A request using the HTTP "DIVINE" method and the
"application/json"
type <bcp14>MUST</bcp14> be sent to the fortune heuristic endpoint to
retrieve
results for a fortune heuristic query.</t>
<t>The request made <bcp14>MUST</bcp14> conform to the specifications of
the
endpoint, as retrieved via the "SEEK" method described in
<xref target= "endpoint-specification-request"/>.</t>
<t>An example of a request is provided below.</t>
<figure>
<sourcecode><![CDATA[
URI: /divination/astrology
Method: DIVINE
Content-Type: application/json
Content-Language: en

{
```

```
"birthday": "1976-02-11T00:00:00Z",
"targetDateBegin": "2018-01-01T00:00:00Z",
"targetDateEnd": "2019-01-01T00:00:00Z",
"interval": "M"
}
]]></sourcecode>
</figure></section>
<section anchor= "endpoint-report-response" numbered=
"false"><name>Fortune Heuristics Report Response</name><t>A fortune
heuristic query using the "DIVINE" method triggers a
response that contains a fortune heuristics report.</t>
<t>A successful response returns a JSON object that <bcp14>MUST</bcp14>
conform
to the object structure described in this section.</t>
<t>A report object <bcp14>SHOULD</bcp14> contain the following
members:</t>
<dl>
<dt>
<tt>type</tt>
</dt>
<dd>(URI, mandatory) A URI that defines the type of the report located
at the <tt>report</tt> key of this object.</dd>
<dt>
<tt>report</tt>
</dt>
<dd>(object, mandatory) An object that contains two keys,
<tt>intervals</tt>
and <tt>events</tt>. The <tt>intervals</tt> object contains an array of
interval
objects that matches the demanded intervals in the request within
the target date range.
The <tt>events</tt> object contains an array of significant event
objects
within the target date range.</dd>
</dl>
<t>An example of a response is provided below.</t>
<figure>
<sourcecode><![CDATA[
URI: /divination/astrology
Method: DIVINE
HTTP/1.1 200 Success
Content-Type: application/json
Content-Language: en

{
"type": "https://association-of.astrology/reports/monthly",
"report": {
"intervals": [
```

```
{
"dateStart": "2018-01-01T00:00:00Z",
"dateEnd": "2018-02-01T00:00:00Z",
"categories": [
{
"category":
"https://divine.example.com/astrology/categories/health"
"value": 80,
"description": "Charge ahead with excellent health."
},
{
"category":
"https://divine.example.com/astrology/categories/love"
"value": 70,
"description":
"Give a certain person or situation another try!"
},
{
"category":
"https://divine.example.com/astrology/categories/finance"
"value": 5,
"description": "You've just realized that you don't have
any cash on hand."
}
]
},
{
"dateStart": "2018-02-01T00:00:00Z",
"dateEnd": "2018-03-01T00:00:00Z",
"..."
},
"..."
],
"events": [
{
"dateStart": "2018-01-15T03:20:00",
"dateEnd": "2018-01-16T20:22:15",
"description": "The planet of growth and good luck, Jupiter
will make a harmonious connection with power planet Pluto,
helping you connect with influential people",
"recommendation": "Engage in networking during this time."
},
{
"dateStart": "2018-03-22T00:12:40",
"dateEnd": "2018-03-28T02:45:03",
"description": "Communication planet Mercury enters your sign,
which will find you in a busier and chattier mood.",
"recommendation":
```

```
"Take charge of work with your newfound energy."
}
"..."
]
}
}
]]></sourcecode>
</figure>
<t>Fortune heuristic reports are created by a divination output that
<bcp14>MAY</bcp14> requires quantitative interpretation. A sample
representation of
interpreting a graphical divination output is provided in
<xref target= "divination-message"/>.</t>
<figure anchor= "divination-message">
<name>Forty-nine yarrow sticks reveals a mystical message on
fortune</name>
<artwork type= "ascii-art" alt= "A mystical pattern in
ASCII"><![CDATA[
                                    0000000000000000000000000
        0000000000000000000000001 G 1000000000000000000000000
        0000000000000000000000011 R 1100000000000000000000000
        0000000000000000000000111 A 1110000000000000000000000
        0000000000000000000001111 C 1111000000000000000000000
        0000000000000000000011111 E 1111100000000000000000000
        0000000000000000000111111 , 1111110000000000000000000
        0000000000000000001111111   1111111000000000000000000
        0000000011111111111111111 M 1111111111111111100000000
        0000000111111111111111111 E 1111111111111111110000000
        0000001111111111111111111 R 1111111111111111111000000
        0000011111111111111111111 C 1111111111111111111100000
        0000111111111111111111111 Y 1111111111111111111110000
        0001111111111111111111111 , 1111111111111111111111000
        0011111111111111111111111   1111111111111111111111100
        0111111111111111111111111 A 1111111111111111111111110
        0000000000000000011111111 N 1111111100000000000000000
        0000000000000000111111111 D 1111111110000000000000000
        0000000000000001111111111   1111111111000000000000000
        0000000000000011111111111 P 1111111111100000000000000
        0000000000000111111111111 E 1111111111110000000000000
        0000000000001111111111111 A 1111111111111000000000000
        0000000000011111111111111 C 1111111111111100000000000
        0000000000111111111111111 E 1111111111111110000000000
        0000000001111111111111111 . 1111111111111111000000000
]]></artwork>
</figure></section>
<section anchor= "endpoint-report-interval-obj" numbered=
"false"><name>Report Interval Object</name><t>The <tt>intervals</tt>
value of a report object contains a number of report
```

```
intervals &#8212; each representing a non-overlapping period
of the
selected interval length. When all of these intervals are put
together, the combined period <bcp14>MUST</bcp14> fully cover the
requested
report target period.</t>
<t>An example interval object is shown below.</t>
<figure>
<sourcecode type= "json"><![CDATA[
{
"dateStart": "2018-01-01T00:00:00Z",
"dateEnd": "2018-02-01T00:00:00Z",
"categories": [
{
"category":
"https://divine.example.com/astrology/categories/health"
"value": 80,
"description": "Charge ahead with your excellent health."
},
{
"category":
"https://divine.example.com/astrology/categories/love"
"value": 70,
"description": "Give a certain person or situation another try!"
},
{
"category":
"https://divine.example.com/astrology/categories/finance"
"value": 5,
"description": "You've just realized that you don't have
any cash on hand."
}
]
}
]]></sourcecode>
</figure>
<t>An interval object <bcp14>MUST</bcp14> contain the following
members:</t>
<dl>
<dt>
<tt>dateStart</tt>
</dt>
<dd>(datetime, mandatory) This value specifies the start of the period
which this interval object applies to.</dd>
<dt>
<tt>dateEnd</tt>
</dt>
<dd>(datetime, mandatory) This value specifies the end of the period
```

```
which this interval object applies to.</dd>
</dl>
<t>In the given example, the <tt>categories</tt> key is an
implementation
specific object that details heuristic results returned by the
selected algorithm. This <bcp14>MAY</bcp14> differ in different
algorithms.</t></section>
<section anchor= "endpoint-report-event-obj" numbered=
"false"><name>Report Events Object</name><t>The <tt>events</tt> value of
a report object contains a number of event
objects. Each event object represents an event relevant to the
calculation of fortune heuristics during a target report period. These
events <bcp14>MAY</bcp14> be of variable time lengths, and
<bcp14>MAY</bcp14> be overlapping
amongst each other.</t>
<t>The following example demonstrates two event objects the service
determines relevant to a user&#8217;s query.</t>
<figure>
<sourcecode type= "json"><![CDATA[
{
"dateStart": "2018-01-15T03:20:00",
"dateEnd": "2018-01-16T20:22:15",
"description": "The planet of growth and good luck, Jupiter will
make a harmonious connection with power planet Pluto, helping you
connect with influential people",
"recommendation": "Engage in networking during this time."
},
{
"dateStart": "2018-03-22T00:12:40",
"dateEnd": "2018-03-28T02:45:03",
"description": "Communication planet Mercury enters your sign,
which will find you in a busier and chattier mood.",
"recommendation": "Take charge of work with your newfound energy."
}
]]></sourcecode>
</figure>
<t>Similar to an interval object, an event object <bcp14>MUST</bcp14>
contain the
following members:</t>
<dl>
<dt>
<tt>dateStart</tt>
</dt>
<dd>(datetime, mandatory) This value specifies the start of the period
described by the event.</dd>
<dt>
<tt>dateEnd</tt>
</dt>
```

```
<dd>(datetime, mandatory) This value specifies the end of the period
described by the event.</dd>
</dl>
<t>In the given example, the keys <tt>description</tt> and
<tt>recommendation</tt>
are implementation-specific details. This <bcp14>MAY</bcp14> differ in
different
algorithms.</t></section>
<section anchor= "endpoint-report-errors" numbered= "false"><name>Report
Generation Errors</name><t>This specification makes use of normal HTTP
error codes with the
following extensions.</t>
<t>Errors <bcp14>MUST</bcp14> be returned using the Problem JSON
Structure as
accordance with <xref target= "RFC7807"/>.</t>
<dl>
<dt>422 Unprocessable Entity</dt>
<dd>For example, a malformed date-time parameter, or an illogical
input,
such as when the subject&#8217;s birthday occurs after the report target
date period.</dd>
<dt>473 Beyond Existing Capability</dt>
<dd>The service determines that the outcome is too difficult to
predict.
For example, in the case where the calculation is too complex to
complete in a certain time period. The service <bcp14>SHOULD</bcp14>
issue this
response code to indicate that the client should not try the same
request again.</dd>
<dt>474 Outcome Impossible</dt>
<dd>The service determines that the outcome is impossible. For
example,
when the algorithm determines that the subject will have deceased
before the start of the requested target period.</dd>
</dl></section></section>
<section anchor= "security" numbered= "false">
<name>Security Considerations</name>
<ul>
<li>TLS <xref target= "RFC5246"/> and authenticated HTTP requests
should be used to
protect the DIVINE request and responses due to the personal nature
of information transmitted.</li>
<li>A client <bcp14>SHOULD</bcp14> verify the identity of the server
on every
request to prevent impersonation or man-in-the-middle attacks, as data
transmitted to and from the server is sensitive information, and at
times critical information to the user.</li>
<li>Synchronization of client and server time <bcp14>MUST</bcp14> be
```

```
well-considered in the implementation of this specification. A
mismatch of client and server time <bcp14>MAY</bcp14> lead to algorithm
miscalculations that can cause mistaken choices of a user that depends
on the reliability of this system.</li>
</ul>
</section>
<section anchor= "iana" numbered= "false">
<name>IANA Considerations</name>
<section anchor= "_well_known_uri_registrations" numbered=
"false"><name>Well-Known URI Registrations</name><t>This document
defines a well-known URI using the registration
procedure and template from <relref section= "5.1" displayFormat= "of"
target="RFC5785"/>.</t>
<section anchor= "_fortune_well_known_uri_registration" numbered=
"false"><name>"fortune" Well-Known URI Registration</name><dl>
<dt>URI suffix</dt>
<dd>fortune</dd>
<dt>Change controller</dt>
<dd>IETF</dd>
<dt>Specification document(s)</dt>
<dd>This document</dd>
<dt>Related information</dt>
<dd>N/A.</dd>
</dl>


<!-- tag::sample[] -->

</section></section>
</section>
</middle><back>
<references anchor= "_normative_references">
<name>Normative References</name>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.2119.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.5785.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7230.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7234.xml"
```

```
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.7807.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8174.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.8259.xml"
parse= "text"/>
</references>
<references anchor= "_informative_references">
<name>Informative References</name>
<reference anchor= "ISO.8601-1.2018" target=
"https://www.iso.org/en/standard/70907.html">
<front>
<title>ISO/DIS 8601-1:2018, Data elements and interchange
formats -- Information interchange -- Representation of dates
and times -- Part 1: Basic rules</title>
<author>
<organization>ISO/IEC</organization>
<address>
<uri>http://www.iso.org</uri>
</address>
</author>
<date month= "January" year= "2018"/>
<abstract><t/></abstract>
</front>
</reference>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.3339.xml"
parse= "text"/>
<xi:include href=
"https://xml2rfc.tools.ietf.org/public/rfc/bibxml/\
reference.RFC.5246.xml"
parse= "text"/>
</references>
<section anchor= "_acknowledgements" numbered=
"false"><name>Acknowledgements</name><t>The authors thank the following
individuals for their valuable
feedback on this specification, and commend them for making fortune
heuristics more accessible for the benefit of mankind.</t>
```

```
<!-- end::sample[] -->

</section>
</back>
</rfc>
<CODE ENDS>
```

Appendix B. Acknowledgements

The authors would like to thank the following persons for their valuable advice and input.

- Adrian Farrel for his comprehensive review of the document and numerous beneficial suggestions.

Authors' Addresses

Ronald Henry Tse
Ribose
Suite 1111, 1 Pedder Street
Central, Hong Kong
Hong Kong

Email: ronald.tse@ribose.com
URI: https://www.ribose.com

Nick Nicholas
Ribose
Australia

Email: nick.nicholas@ribose.com
URI: https://www.ribose.com

Jeffrey Lau
Ribose
Suite 1111, 1 Pedder Street
Central, Hong Kong
Hong Kong

Email: jeffrey.lau@ribose.com
URI: https://www.ribose.com

Paolo Brasolin
Ribose
Italy

Email: paolo.brasolin@ribose.com
URI: https://www.ribose.com