

Independent Submission
Request for Comments: 7328
Category: Informational
ISSN: 2070-1721

R. Gieben
Google
August 2014

Writing I-Ds and RFCs Using Pandoc and a Bit of XML

Abstract

This document presents a technique for using a Markdown syntax variant, called Pandoc, and a bit of XML (as defined in RFC 2629) as a source format for documents that are Internet-Drafts (I-Ds) or RFCs.

The goal of this technique (which is called Pandoc2rfc) is to let an author of an I-D focus on the main body of text without being distracted too much by XML tags; however, it does not alleviate the need to typeset some files in XML.

Status of This Memo

This document is not an Internet Standards Track specification; it is published for informational purposes.

This is a contribution to the RFC Series, independently of any other RFC stream. The RFC Editor has chosen to publish this document at its discretion and makes no statement about its value for implementation or deployment. Documents approved for publication by the RFC Editor are not a candidate for any level of Internet Standard; see Section 2 of RFC 5741.

Information about the current status of this document, any errata, and how to provide feedback on it may be obtained at <http://www.rfc-editor.org/info/rfc7328>.

Copyright Notice

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

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document.

Table of Contents

1. Introduction	2
2. Pandoc to RFC	2
2.1. Dependencies	5
3. Building an Internet-Draft	5
4. Supported Features	5
5. Unsupported Features and Limitations	7
6. Pandoc Style	7
6.1. Figures	7
6.2. Tables	7
6.3. References	7
6.4. Index	8
7. Acknowledgements	8
8. Security Considerations	8
9. References	8
9.1. Normative References	8
9.2. Informative References	9
Appendix A. Cheat Sheet	10

1. Introduction

This document presents a technique for using a Markdown [Markdown] syntax variant, called Pandoc [Pandoc], and a bit of XML [RFC2629] as a source format for documents that are Internet-Drafts (I-Ds) or RFCs.

The goal of this technique is to let an author of an I-D focus on the main body of text without being distracted too much by XML tags; however, it does not alleviate the need to typeset some files in XML.

Pandoc is a format that is almost plain text and therefore particularly well suited for editing RFC-like documents. The syntax itself is a superset of the syntax championed by Markdown.

2. Pandoc to RFC

Pandoc's syntax is easy to learn and write, and it can be translated to numerous output formats, including, but not limited to: HTML, EPUB, (plain) Markdown, and DocBook XML.

Pandoc2rfc allows authors to write in Pandoc syntax that is then transformed to XML and given to xml2rfc. The conversions are, in a way, amusing, as we start off with (almost) plain text, use elaborate XML, and end up with plain text again.

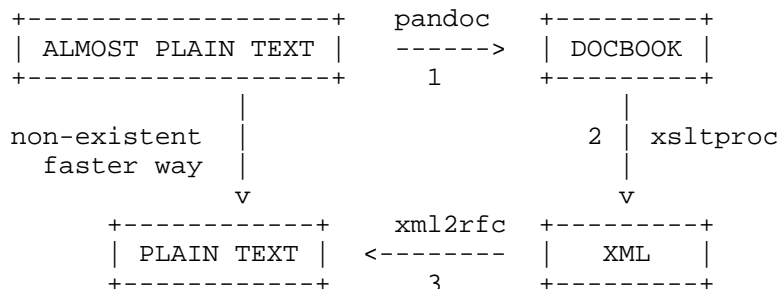


Figure 1: Attempt to justify Pandoc2rfc

The output of step 2 in Figure 1 is XML that is suitable for inclusion in either the "middle" or "back" section of an RFC.

Even though Pandoc2rfc abstracts away a lot of XML details, there are still places left where XML files needs to be edited -- most notably in the "front" section of an RFC.

The simplest way to start using Pandoc2rfc is to create a template XML file and include the appropriate XML for the "front", "middle", and "back" section:

```
<?xml version='1.0' ?>
<!DOCTYPE rfc SYSTEM 'rfc2629.dtd' [
<!ENTITY pandocAbstract PUBLIC '' 'abstract.xml'>
<!ENTITY pandocMiddle PUBLIC '' 'middle.xml'>
<!ENTITY pandocBack PUBLIC '' 'back.xml'>
<!ENTITY rfc.2629 PUBLIC '' 'reference.RFC.2629.xml'>
]>

<rfc ipr='trust200902' docName='draft-string-example'>
  <front>
    <title>Writing I-Ds and RFCs using Pandoc</title>
    <author>
      <organization/>
      <address><uri>http://www.example.com</uri></address>
    </author>
    <date/>
    <abstract>
      &pandocAbstract;
    </abstract>
  </front>
  <middle>
    &pandocMiddle;
  </middle>
  <back>
    <references title="Normative References">
      &rfc.2629;
    </references>
    &pandocBack;
  </back>
</rfc>
```

Figure 2: A minimal template.xml

In this case, you will need to edit four documents:

1. "abstract.mkd" - contains the abstract;
2. "middle.mkd" - contains the main body of text;
3. "back.mkd" - holds the appendices (if any);
4. and this "template.xml" -- probably a fairly static file; among other things, it holds the author(s) and the references.

Up-to-date source code for Pandoc2rfc can be found at [Pandoc2rfc]; this includes the style sheet "transform.xsl", which is used for the XML transformation (also see Section 3).

2.1. Dependencies

Pandoc2rfc needs "xsltproc" [XSLT] and "pandoc" [Pandoc] to be installed. The conversion to xml2rfc XML is done with a style sheet based on XSLT version 1.0 [W3C.REC-xslt-19991116].

When using the template from Figure 2, xml2rfc version 2 (or higher) must be used.

3. Building an Internet-Draft

Assuming the setup from Section 2, we can build an I-D as follows (in a Unix-like environment):

```
for i in abstract middle back; do
pandoc -st docbook $i.mkd | xsltproc --nonet transform.xsl - > $i.xml
done

xml2rfc template.xml -f draft.txt --text # create text output
xml2rfc template.xml -f draft.html --html # or create HTML output
xml2rfc template.xml -f draft.xml --exp  # or create XML output
```

Figure 3: Building an I-D

Note that the output file names (abstract.xml, middle.xml, and back.xml) must match the names used as the XML entities in "template.xml". (See the "!ENTITY" lines in Figure 2.) The Pandoc2rfc source repository includes a shell script that incorporates the above transformations. Creating a "draft.txt" or a "draft.xml" can be done with "pandoc2rfc *.mkd" and "pandoc2rfc -X *.mkd", respectively.

4. Supported Features

The full description of Pandoc's syntax can be found in [PandocGuide]. The following features of xml2rfc are supported by Pandoc2rfc (also see Table 1 in Appendix A for a "cheat sheet"):

- o Sections with an anchor and title attributes;
- o Several list styles:
 - * style="symbols", use "*" for each item;
 - * style="numbers", use digits: "1. " for each item;
 - * style="empty", use "#. " for each item;

- * style="format %i", use lowercase Roman numerals: "ii. ";
- * style="format (%d)", use uppercase Roman numerals "II. ";
- * style="format ...", use strike-through text at the start in the first element, "1. ~~REQ%d.~~ ";
- * style="letters", use lower- or uppercase letters: "a. " and "A. " (note: two spaces as mandated by Pandoc);
- * style="hanging", use the Pandoc definition list syntax:
 - Term 1
 - : Definition 1
- o Spanx style="verb", style="emph", and style="strong", respectively, use: "`text`", "_text_" or "***text***";
- o Block quote, which is converted to a paragraph within a "<list style='empty'>";
- o Figures with an anchor and title (Section 6.1);
- o Tables with an anchor and title (Section 6.2);
- o References (Section 6.3)
 - * external ("<eref>");
 - * cross-reference ("<xref>"), to:
 - + sections (handled by Pandoc);
 - + figures (handled by XSLT);
 - + tables (handled by XSLT).
- o Index, by using footnotes and superscript text (Section 6.4);
- o Citations, by using cross-references;
- o Processing Instructions (PIs), which appear as "<?rfc?>", may be used after a section header. They are carried over to the generated XML.
- o The "<vspace>" tag is supported and carried over to the generated XML.

5. Unsupported Features and Limitations

With Pandoc2rfc, an author of an I-D can get a long way without needing to input XML, but it is not a 100% solution. The initial setup and the reference library still force the author to edit XML files. The metadata feature (Pandoc's "Title Block" extension) is not used in Pandoc2rfc. This information (authors, date, keyword, and URLs) should be put in the "template.xml".

Some other quirks:

- o Comments are supported via HTML comments in the Pandoc source files.
- o Citations are supported via cross-references; the citation syntax of Pandoc is not used.
- o Authors still need to know how to deal with possible errors from xml2rfc.

6. Pandoc Style

The following sections detail how to use the Pandoc syntax for figures, tables, and references to get the desired output.

6.1. Figures

Indent the paragraph with 4 spaces as mandated by Pandoc. If you add an inline footnote `_directly_` after the figure, the artwork gets a title attribute with the text of that footnote (and a possible anchor).

6.2. Tables

A table can be entered by using Pandoc's table syntax. You can choose multiple styles as input, but they all are converted to the same style table (plain "`<texttable>`") in xml2rfc. If you add an inline footnote `_directly_` after the table, it will get a title attribute with the text of that footnote (and a possible anchor). The built-in syntax of Pandoc to create a caption with "Table:" should not be used.

6.3. References

Pandoc provides a syntax that can be used for references. Its syntax is repeated in this paragraph. Any reference like "[Click here](URI)" is an external reference. An internal reference (i.e., "see Section X") is typeset with "[](#localid)".

For referencing RFCs (and other documents), you will need to add the reference source in the template as an external XML entity; Figure 2 provides an example. After that, you can use the following syntax to create a citation: "[](#RFC2629)" to cite RFC 2629.

There is no direct support for referencing tables, figures, and artworks, but Pandoc2rfc employs the following "hack". If an inline footnote is added after the figure or table, the text of the footnote is used as the title. The first word up until a double colon "::" will be used as the anchor. If a figure has an anchor, it will be centered on the page.

Figure 2, for instance, is followed by this inline footnote:

```
^[fig:minimal::A minimal template.xml.]
```

6.4. Index

An index can be generated by using the following syntax:

```
^[ ^item^ subitem ]
```

where "subitem" is optional.

7. Acknowledgements

The following people have helped shape Pandoc2rfc: Benno Overeinder, Erlend Hamnaberg, Matthijs Mekking, and Trygve Laugstoel.

8. Security Considerations

This document raises no security issues.

9. References

9.1. Normative References

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

[W3C.REC-xslt-19991116]
Clark, J., "XSL Transformations (XSLT) Version 1.0", World Wide Web Consortium Recommendation REC-xslt-19991116, November 1999,
<<http://www.w3.org/TR/1999/REC-xslt-19991116>>.

[XSLT] Veillard, D., "The XSLT C library for GNOME", 2006,
<<http://xmlsoft.org/XSLT/xsltproc2.html>>.

9.2. Informative References

- [Markdown] Gruber, J., "Markdown", 2004,
<<http://daringfireball.net/projects/markdown/>>.
- [Pandoc] MacFarlane, J., "Pandoc, a universal document converter",
2006, <<http://johnmacfarlane.net/pandoc/>>.
- [Pandoc2rfc]
Gieben, R., "Pandoc2rfc git repository", October 2012,
<<http://github.com/miekg/pandoc2rfc>>.
- [PandocGuide]
MacFarlane, J., "Pandoc User's Guide", 2006,
<<http://johnmacfarlane.net/pandoc/README.html>>.

Appendix A. Cheat Sheet

Textual construct	Pandoc syntax	Text output
Section Header	"# Section"	1. Section
Unordered List	"* item"	o item
Unordered List	"#. item"	item
Ordered List	"1. item"	1. item
Ordered List	"a. item"	a. item
Ordered List	"ii. item"	i. item
Ordered List	"II. item"	(1) item
Ordered List	"A. item"	A. item
Ordered List	"1. ~~REQ%d.~~"	REQ1.
Emphasis	"_text_"	_text_
Strong Emphasis	"**text**"	*text*
Verbatim	"`text`"	"text"
Block Quote	"> quote"	quote
External Reference	"[Click](URI)"	Click [1]
Internal Reference	"[](#id)"	Section 1
Figure Anchor	"^[fid::text]"	N/A
Figure Reference	"[](#fid)"	Figure 1
Table Anchor	"^[tid::text]"	N/A
Table Reference	"[](#tid)"	Table 1
Citations	"[](#RFC2119)"	[RFC2119]
Table	Tables	*
Figures	Code Blocks	*
Definition List	Definition	*
Index	^[^item^]	*

* This construct creates output too voluminous to show in the table.

Table 1: The most important textual constructs that can be used in Pandoc2rfc

Author's Address

R. (Miek) Gieben
Google

E-Mail: miek@google.com

