

RFC 7749 : The 'XML2RFC' version 2 Vocabulary

Stéphane Bortzmeyer
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 7 février 2016

Date de publication du RFC : Février 2016

<https://www.bortzmeyer.org/7749.html>

Contrairement à beaucoup de SDO, l'IETF n'a pas de format standard pour l'écriture de ses documents. XML est toutefois le plus répandu, avec le vocabulaire qui avait été autrefois décrit dans le RFC 2629¹, et qui est maintenant spécifié dans ce RFC. (Il a depuis été remplacé par le RFC 7991.)

Vous voulez écrire un RFC ? Un des choix possibles est un format XML, fondé sur un vocabulaire spécifique aux RFC, et mis en œuvre dans l'outil `xml2rfc` <<http://xml2rfc.ietf.org/>>. Ce vocabulaire n'avait pas été mis à jour depuis seize ans (alors que l'outil évoluait) et c'est seulement maintenant, après un processus long et douloureux, que les auteurs d'"*Internet-Drafts*" et de RFC disposent enfin d'une description officielle et à jour du format dans lequel ils travaillent. Voici donc le vocabulaire « XML2RFC version 2 ».

Le format principal de publication des RFC est du texte brut, pour des raisons de portabilité, d'indépendance vis-à-vis de tout éditeur de logiciels spécifique, et de facilité de traitement automatique. Mais aucun auteur n'écrit les RFC directement dans ce format (rien que pour la numérotation des pages, cela serait infernal). Ils écrivent en des formats divers, et le convertissent ensuite en texte brut. À noter que certains éléments XML décrits ici ne produisent rien de particulier dans le texte final, mais sont utiles pour d'autres usages, comme l'indexation.

Voici le squelette d'un "*Internet-Draft*" écrit avec ce XML :

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc2629.txt>

```

<?xml version="1.0" encoding="utf-8"?>
<rfc docName="draft-ietf-dnsop-qname-minimisation-09"
  category="exp" ipr="trust200902">
<front>
<title abbrev="Qname minimisation">DNS query name minimisation to improve privacy</title>
...
<middle>
<section anchor="intro" title="Introduction and background">
<t>The problem statement is described in <xref
target="RFC7626"/>. [...]
...
</back>
</rfc>

```

(Un squelette plus détaillé est disponible en <https://www.rfc-editor.org/materials/template-bare-06.txt>.) Sur ce squelette simple, on voit l'élément racine (`<rfc>`), l'utilisation des attributs (comme `category` qui indique le statut du futur RFC, ici « expérimental »), la séparation en trois parties, `<front>`, qui regroupe les métadonnées, `<middle>`, qui est le texte principal, et `<back>`, où se trouvent la bibliographie, les annexes, etc.

Parmi les attributs de cet élément racine `<rfc>`, notez `ipr`, qui indique les conditions légales d'utilisation de ce RFC. Dans cet exemple, la valeur est la plus couramment utilisée : `trust200902` (cf. l'annexe A.2) indique les règles de l'IETF Trust <https://trustee.ietf.org/> datant de 2009 (qui disent en gros que le texte du RFC peut être librement copié, reproduit, distribué et mis en œuvre dans des programmes). L'annexe A de notre RFC détaille ce qu'on appelle le "boilerplate", ces textes juridiques obligatoires qui sont ajoutés automatiquement par le logiciel `xml2rfc`. Ainsi, si on met `ipr="trust200902"` dans l'élément `<rfc>`, `xml2rfc` va automatiquement ajouter « *Copyright (c) 2015 IETF Trust and the persons identified as the document authors. All rights reserved. \ This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents [...]* »...

Le gros morceau du RFC est la section 2, qui donne la liste des éléments XML acceptés. Je ne vais pas reproduire ici cette liste, juste parler de quelques éléments qui me semblent rigolos.

`<section>` contient une partie du RFC. Cet élément est hiérarchique : on crée des sous-sections en les mettant sous les sections existantes, et ainsi de suite, récursivement. (Contrairement à ce qui se passe avec HTML, où on indique explicitement le niveau de la section, `<h1>`, `<h2>`, etc.)

`<t>` contient un paragraphe et est donc l'équivalent du `<p>` de HTML.

`<artwork>` est le seul élément qui permet de spécifier du texte qui sera représenté comme tel, sans aucune justification, mise à la ligne, etc. Il a gagné plusieurs attributs par rapport au RFC 2629, pour bien contrôler le résultat. `<artwork>` permet d'inclure du code source ou bien de mettre de l'art ASCII dans un RFC (pour l'instant, il n'y a pas encore d'autre mécanisme simple pour des images, mais c'est quand même possible, cf. le RFC 5598 avec ses différentes versions <http://www.rfc-editor.org/info/rfc5598>). Notez que l'attribut `src` permet de spécifier un fichier externe, l'art ASCII ne servant alors que de solution de secours, pour le format en texte seul. Voici un exemple :

```

<figure title="A network">
<artwork>
+-----+ +-----+
| Alice |-----| Bob |

```

<https://www.bortzmeyer.org/7749.html>

```
| 2001:db8::1 | | 2001:db8::2 |
+-----+ +-----+
</artwork>
</figure>
```

`<eref>` permet de faire un lien hypertexte vers l'extérieur :

```
<t>More text and a <eref
target="http://www.rfc-editor.org/">lien vers le site du RFC Editor</eref>.</t>
```

`<list>` permet de représenter les traditionnelles listes à puces :

```
<t>There are three sorts of DNS requests being issued:
<list>
<t>Primary request: [...]</t>
<t>Secondary requests: [...]</t>
<t>Tertiary requests: [...]</t>
</list>
</t>
```

`<references>` permet d'indiquer une bibliographie. Il y en a typiquement deux dans un RFC (cf. la section 4.8.6 du RFC 7322), la bibliographie normative (ce qu'il faut absolument avoir lu et compris car le RFC en dépend) et l'informatrice (ce qu'on peut sauter si on est pressé). Pour aider, le "*RFC Editor*" distribue des fichiers XML contenant les références aux RFC publiés, comme `<http://www.rfc-editor.org/refs/bibxml/reference.RFC.7626.xml>`.

Si vous voulez un exemple plus complet, regardez ce fichier (en ligne sur <https://www.bortzmeyer.org/files/test-xml2rfc-v2.xml>). Vous pouvez le traiter avec `xml2rfc` :

```
% xml2rfc test-xml2rfc-v2.xml
[...]
Created file test-xml2rfc-v2.txt
```

Et vous obtenez un joli RFC.

Ce format de RFC s'appuie sur XML et il faut donc suivre les règles de XML, notamment sur les caractères spéciaux. Ainsi, le chevron ouvrant doit être remplacé par une séquence d'échappement (`<` au lieu de `<`). Si cette contrainte est trop forte, on peut aussi enclore les parties à « échapper » dans une section CDATA.

Le format des RFC ne permet pas d'autres caractères que ceux du jeu ASCII. Néanmoins, certains caractères Unicode peuvent être utiles dans le source XML (section 4), pour contrôler certains aspects de la présentation :

<https://www.bortzmeyer.org/7749.html>

- Espace insécable U+00A0,
- Tiret insécable U+2011,
- Gluon de mots (empêche la césure entre deux lettres) U+2060.

La restriction actuelle à ASCII (elle était dans le RFC 2223, section 3a, mais n'a pas été formellement reprise dans son successeur, le RFC 7322) sera peut-être levée dans le futur, et ce RFC devra alors être mis à jour (section 6).

Le format actuel ne permet pas l'inclusion d'autres documents, donc un RFC doit être en une seule pièce, un seul fichier XML (section 5). On peut toutefois utiliser les mécanismes génériques d'inclusion de XML, et c'est souvent utilisé pour la bibliographie :

```
<!DOCTYPE rfc [
  <!ENTITY rfc7830 PUBLIC
    "http://xml.resource.org/public/rfc/bibxml/reference.RFC.7830.xml">
]>

[...]
<references>
  &rfc7830;
</references>
```

À noter qu'il existe désormais un type MIME pour les sources XML de RFC, `application/rfc+xml` (section 8 de notre RFC).

Si vous voulez voir le schéma XML complet, il est en annexe C (j'en ai exporté une version utilisable telle quelle, sans les sauts de page des RFC, en (en ligne sur <https://www.bortzmeyer.org/files/rfc-schema.rnc>)). Comme il est écrit en Relax NG, il permet l'utilisation de tous les outils Relax NG, comme le mode emacs `nxml-mode` (<https://www.emacswiki.org/emacs/NxmlMode>) et comme `rnv` (<http://www.davidashen.net/rnv.html>). Ainsi, une fois le fichier `rfc-schema.rnc` chargé dans emacs (menus XML puis "Set schema" puis "File"), on dispose de fonctions d'édition bien pratiques (par exemple, on tape un `;` puis une tabulation et emacs propose de compléter uniquement avec les éléments autorisés à cet endroit). Cela évite bien des erreurs.

À noter que ce RFC ne décrit que les éléments et attributs XML, pas les "processing instructions" (PI) `<?rfc ... ?>` qui servent, par exemple, à contrôler si une table des matières est affichée. Pour connaître celles acceptées par `xml2rfc`, il faut regarder la documentation de l'outil (<http://svn.tools.ietf.org/svn/tools/xml2rfc/archive/README.html>) et la FAQ (http://xml2rfc.ietf.org/xml2rfcFAQ.html#q_pis).

Avec un logiciel comme `rnv` (<http://www.davidashen.net/rnv.html>), on peut tester la syntaxe (uniquement la syntaxe : certaines contraintes dans le RFC ne sont pas exprimables dans le schéma, il a fallu les formuler en langue naturelle dans le texte du RFC) :

```
% rnv rfc.rnc test-xml2rfc-v2.xml
test-xml2rfc-v2.xml
```

Parfait, ici, tout est bon. S'il y avait eu une erreur :

<https://www.bortzmeyer.org/7749.html>

```
% rnv rfc.rnc test-xml2rfc-v2.xml
test-xml2rfc-v2.xml
test-xml2rfc-v2.xml:9:4: error: element ^t not allowed
required:
element ^section
```

Si le RFC contient des références externes (que rnv ne sait pas traiter), on peut utiliser xmllint pour les remplacer :

```
% xmllint --dropdtd --noent draft-ietf-dnsop-nxdomain-cut.xml | rnv rfc.rnc
```

À noter que le RFC n'utilise pas rnv mais Jing <<http://www.thaiopensource.com/relaxng/jing.html>> (annexe C.1). Mode d'emploi très court, on télécharge <<https://code.google.com/archive/p/jing-trang/downloads>> :

```
% https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/jing-trang/jing-20091111.zip
% unzip jing-20091111.zip
% java -jar ./jing-20091111/bin/jing.jar -c rfc.rnc draft-ietf-dnsop-nxdomain-cut.xml
%
```

Les changements depuis le texte précédent, le RFC 2629, seize ans auparavant, sont résumés dans l'annexe B. L'élément <appendix>, qui servait pour les annexes des RFC (annexes assez fréquentes, de nos jours), a été supprimé. Pour faire une annexe, désormais, on met une <section> dans le <back>. En revanche, sept nouveaux éléments ont été ajoutés dont les plus importants, à mon avis, sont <texttable>, <ttcol> et <c>, qui permettent enfin de faire des tableaux proprement dans les RFC.