

# RFC 7991 : The "xml2rfc" Version 3 Vocabulary

Stéphane Bortzmeyer  
<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 16 décembre 2016

Date de publication du RFC : Décembre 2016

<https://www.bortzmeyer.org/7991.html>

---

Contrairement à beaucoup de SDO, l'IETF n'avait pas de format standard pour l'écriture de ses documents. Désormais, avec le nouveau cadre décrit dans le RFC 7990<sup>1</sup>, c'est fait. XML, avec le vocabulaire décrit dans ce nouveau RFC, est le format canonique des RFC.

Vous voulez écrire un RFC? Il est fortement recommandé d'utiliser dès le début le format standard XML, fondé sur un vocabulaire spécifique aux RFC, et mis en œuvre dans la future version de l'outil xml2rfc <<http://xml2rfc.ietf.org/>>. Voici donc le vocabulaire « XML2RFC version 3 », succédant à deux versions qui n'étaient pas officiellement standard (les changements depuis la v2, spécifiée dans le RFC 7749, ne sont pas énormes). Notez que le vocabulaire XML et les outils continuent à évoluer, donc ce RFC n'est pas éternel. Et que la version 2 restera sans doute en service pendant encore des années : cela prend du temps de changer les habitudes !

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

```
<?xml version="1.0" encoding="utf-8"?>
<rfc docName="draft-ietf-dnsop-qname-minimisation-09" submissionType="IETF"
  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>
```

---

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7990.txt>

Sur ce squelette simple, on voit l'élément racine (`<rfc>`), l'utilisation des attributs (comme `submissionType` qui indique la voie prise par le document, ici, l'IETF, cf. RFC 7841), 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.1) 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.) On a aussi `<abstract>`, qui indique le résumé au début du RFC.

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

`<artwork>` permet de spécifier du texte qui sera représenté comme tel, sans aucune justification, mise à la ligne, etc. (Les tabulations sont interdites, ce qui règle le vieux débat « *tabs vs. spaces* » `<https://opensourcehacker.com/2012/05/13/never-use-hard-tabs/>` ».) `<artwork>` permet de mettre de l'art ASCII dans un RFC, mais la méthode préférée pour les images est désormais SVG, voir le RFC 7996. Le SVG peut être mis directement dans le source XML ou bien inclus par différentes méthodes, dont l'attribut `src`. Cet 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. Un attribut `type` permet d'indiquer le type du dessin (par exemple `svg` pour les images en SVG). La liste des types possibles sera en ligne. Voici un exemple d'art ASCII :

```
<artwork type="ascii-art">
+-----+ +-----+
| Alice |-----| Bob |
| 2001:db8::1 | | 2001:db8::2 |
+-----+ +-----+
</artwork>
```

Le code source, lui, se fait avec l'élément `<sourcecode>`, un attribut `type` permettant d'indiquer le langage utilisé (une liste des valeurs possibles sera en ligne). Voici un exemple :

```
<sourcecode type="python">
print("Hello, world")
</sourcecode>
```

Comme le langage utilisé peut utiliser des caractères qui sont spéciaux pour XML (comme ; ou &), il est souvent préférable de mettre le code source dans une section CDATA.

`<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>
```

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

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

`<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'informative (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>`.

Le nom d'un auteur de RFC se met avec l'attribut `<author>`. Comme il peut être en caractères non-ASCII, des attributs permettent d'indiquer une variante en ASCII seul. Par exemple :

```
<author fullname="Patrik Fältström" asciiFullname="Patrik Faltstrom">  
<organization>Netnod</organization>  
</author>
```

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 (`&lt;` 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 permet d'autres caractères que ceux du jeu ASCII, mais avec certaines restrictions (voir RFC 7997).

Le format actuel permet l'inclusion d'autres documents, via des attributs comme l'attribut `src` pour le code source :

---

<https://www.bortzmeyer.org/7991.html>

```
<sourcecode type="python" src="hello.py"/>
```

On peut aussi utiliser les mécanismes génériques d'inclusion de XML, comme XInclude (cf. annexe B.1) ou les entités, 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 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-v3.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-v3.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 de "*processing instructions*" (PI), qui ne sont plus acceptées.

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-v3.rnc rfc-v3-sample.xml
rfc-v3-sample.xml
```

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

```
% rnv rfc-v3.rnc rfc-v3-sample-wrong.xml
rfc-v3-sample-wrong.xml
rfc-v3-sample-wrong.xml:9:6: error: element ^t not allowed
required:
element ^section
rfc-v3-sample-wrong.xml:11:2: error: unfinished content of element ^middle
required:
element ^section
error: some documents are invalid
```

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

```
% xmllint --dropdtd --noent draft-dupont-my-protocol.xml | rnv rfc-v3.rnc
```

On peut aussi utiliser Jing <<http://www.thaiopensource.com/relaxng/jing.html>> (annexe C.1). Mode d'emploi très court, on télécharge <<https://github.com/relaxng/jing-trang>> :

```
% wget https://storage.googleapis.com/google-code-archive-downloads/v2/code.google.com/jing-trang/jing-20091111.
% unzip jing-20091111.zip
% java -jar ./jing-20091111/bin/jing.jar -c rfc-v3.rnc draft-dupont-my-protocol.xml
%
```

Les changements depuis le texte précédent, le RFC 7749, qui décrivait la version 2 (notre RFC est la version 3), sont décrits dans l'annexe D et résumés en section 1.3. L'idée était notamment d'avoir un vocabulaire plus facile à utiliser, sans pour autant trop changer par rapport au format v2, qui était bien connu des auteurs.

Le changement le plus spectaculaire concerne les listes, qui sont désormais faites, comme en HTML, avec des <dl>, <ul> et <ol>. Dans une liste, le contenu est marqué par <li> et plus <t>. Autre inspiration HTML, l'apparition des tables, avec <table> (et éléments associés comme <tr> et <td>). D'autre part, de nouveaux éléments apparaissent pour marquer du texte, par exemple s'il est **important** (<em>, qui n'avait pas d'équivalent en v2, dont le seul format de sortie était le texte brut). Il y a aussi un <blockquote> pour les citations. Bien que l'IETF se vante souvent de pratiquer le culte du « *running code* », il n'y avait pas d'élément XML particulier pour indiquer du code source dans un RFC (on se contentait d'<artwork>). C'est désormais fait avec <sourcecode>. Quelques autres éléments XML nouveaux (je ne les cite pas tous, le RFC fait 159 pages!) : <displayreference> pour associer un chouette texte aux références, <link> pour les liens externes (contrairement à <eref>, qui existait déjà, <link> est spécifique à certains types de documents, par exemple les Internet-Drafts) ou encore <br> pour forcer des sauts de ligne (mauvaise idée que de mettre des éléments de présentation, si vous voulez mon avis).

Il y a aussi de nouveaux attributs XML aux éléments existants. Pour remplacer les PI ("*processing instructions*" comme <?rfc toc="yes"?>), on a tocInclude et tocDepth dans l'élément <rfc>, afin de contrôler la table des matières. De même, pour gérer l'internationalisation, il y a désormais un attribut ascii pour les éléments qui acceptent du contenu non-ASCII, afin de fournir une alternative pour les anglophones. Il y a aussi des attributs plus orientés présentation comme keepWithNext ou keepWithPrevious, attributs de <t>, qui expriment un souhait de garder ce paragraphe avec le suivant ou le précédent, pour mieux contrôler la pagination.

En revanche, certains éléments et attributs sont retirés de la circulation. Ils seront encore acceptés par les outils, mais temporairement. <list> subit ce triste sort (remplacé par les éléments HTMLisant comme <ul> et <ol>). <facsimile> disparaît également, victime des évolutions technologiques. Parmi les attributs, title disparaît (il était utilisé dans des éléments comme <section>) au profit de name (changement assez gratuit, je trouve).

Les autres changements sont bien sûr l'acceptation de caractères non-ASCII, et plein de modifications de détail.

Question mise en œuvre, il faudra patienter. S'il y a déjà eu des mises en œuvre expérimentales <<http://greenbytes.de/tech/webdav/rfc2629xslt/rfc2629xslt.html#v3>> et partielles, les vrais logiciels officiels ne sont pas encore, en octobre 2016, développés.