

Utiliser une bibliographie avec Docbook

Stéphane Bortzmeyer

<stephane+blog@bortzmeyer.org>

Première rédaction de cet article le 17 avril 2009

<https://www.bortzmeyer.org/bibliographie-docbook.html>

Le schéma XML de documentation Docbook contient un élément `<bibliography>` qui permet d'indiquer une bibliographie. Gérer le contenu de cette bibliographie à la main dans chaque document Docbook n'est pas pratique : après tout, beaucoup d'éléments de la bibliographie sont certainement partagés entre tous les documents qu'on édite dans son travail. Je présente ici une méthode qui permet de gérer la bibliographie à part, sous forme d'un fichier XML unique, et d'inclure dans les documents Docbook seulement les références effectivement citées.

Le principe est donc d'avoir un seul fichier XML contenant les références bibliographiques utilisées par une personne ou un groupe. Il existe peut-être un schéma standard simple et pratique pour décrire ce fichier mais je ne l'ai pas trouvé et j'ai donc fait un schéma, écrit en DTD et publié en (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-resources.dtd>). Ce schéma permet d'écrire des documents comme :

```
<!DOCTYPE bibliography SYSTEM "internet.dtd">
<bibliography>

  <netresource id="xml-rpc">
    <author>
      <lastname>UserLand Software</lastname>
    </author>
    <title>XML-RPC Home Page</title>
    <year>2003</year>
    <url>http://www.xml-rpc.com/</url>
  </netresource>

  <netresource id="honeynet.knowyourenemy">
    <author><lastname>The Honeynet Project & Research Alliance</lastname></author>
    <title>Know your Enemy: Tracking Botnets</title>
    <year>2005</year>
    <url>http://www.honeynet.org/papers/bots/</url>
  </netresource>

  <book id="iteanu.identite">
    <author><firstname>Olivier</firstname><lastname>Iteanu</lastname></author>
    <title>L'identité numérique en question</title>
```

```

    <year>2008</year>
    <number>978-2-212-12255-8</number>
    <publisher>Eyrolles</publisher>
</book>
...
</bibliography>

```

Ce fichier XML peut être écrit et modifié avec les éditeurs de son choix et géré dans un VCS.

On pourrait bien sûr inclure le fichier entier dans le document Docbook (pas par copier-coller, ce qui empêcherait de profiter des mises à jour, mais par un mécanisme d'inclusion automatique comme XInclude) mais la bibliographie complète peut être énorme. On va donc chercher à extraire l'information utile.

Cela se fait par un programme XSLT, (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-xsl>), qui extrait les éléments comme `<netresource>` si leur identité (l'attribut `id`) est celle indiquée (via le paramètre `refs`). Pour ne pas avoir à indiquer toutes les identités à la main, un autre programme XSLT, (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-find.xsl>), permet de les extraire automatiquement du document Docbook, en regardant les références comme `<xref>`. Voici comment utiliser ces deux programmes en même temps avec le processeur `xsltproc` et le shell Unix (`$ (COMMAND)` signifie « exécuter la commande `COMMAND` et inclure son résultat à cet endroit ») :

```

xsltproc -o SPECIFICBIBLIO.xml \
--stringparam refs "$(xsltproc docbook-biblio-find.xsl MYDOCUMENT.db)" \
    docbook-biblio-extract.xsl MYBIBLIO.xml

```

La commande est longue à taper mais, personnellement, j'automatise cela avec un Makefile.

`docbook-biblio-extract` ne fait qu'extraire les éléments XML, qui sont toujours au format de ma DTD. Pour les traduire en Docbook, un dernier programme XSLT est nécessaire, (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-resources2docbook.xsl>). Il s'utilise ainsi :

```

xsltproc -o SPECIFICBIBLIO.db resources2docbook.xsl SPECIFICBIBLIO.xml

```

Il ne reste plus qu'à mettre ce fichier dans le document Docbook qu'on est en train d'écrire, par exemple via XInclude. Dans ce dernier cas, le document Docbook qu'on écrit contiendra :

```

<xi:include href="SPECIFICBIBLIO.db"/>

```

pourra être traité, par exemple, par `xmllint` :

```

xmllint --xinclude MYDOCUMENT.db > MYDOCUMENT-COMPLET.db

```

Parmi les documents cités dans des articles, les RFC posent un cas particulier. En effet, il existe déjà une liste des RFC en XML (au format décrit dans le RFC 7749¹), il n'y a pas besoin de maintenir cette liste à la main avec les autres. Je crée donc une DTD qui étend la précédente pour les RFC, (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-rfc.dtd>), et un programme pour traduire la liste officielle vers mon format, (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-2629toresources.xsl>). La liste officielle est automatiquement récupérée par `wget http://xml.resource.org/public/rfc/bibxml/index.xml` et traduite dans mon format.

Ensuite, pour finir de traiter les RFC, j'ai des programmes spécifiques, (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-findrfc.xsl>) qui trouve les références de type RFCnnn où NNN est le numéro d'un RFC, et (en ligne sur <https://www.bortzmeyer.org/files/docbook-biblio-extractrfc.xsl>) qui extrait les RFC de la liste officielle.

Toutes ces commandes sont complexes à taper, alors make arrive à la rescousse. Voici le Makefile qui automatise tout cela :

```
rfc-MONDOCUMENT-automatic.xml: rfc.xml MONDOCUMENT.db
    xsltproc -o $@ --stringparam rfc " `xsltproc docbook-biblio-findrfc.xsl MONDOCUMENT.db` " docbook-biblio-
biblio-MONDOCUMENT-automatic.xml: bibliocommune.xml MONDOCUMENT.db
    xsltproc -o $@ --stringparam refs " `xsltproc docbook-biblio-find.xsl MONDOCUMENT.db` " docbook-biblio-ext
MONDOCUMENT-COMPLET.db: MONDOCUMENT.db rfc-MONDOCUMENT-automatic.db biblio-MONDOCUMENT-automatic.db
    xmllint --xinclude $< > $@
%.db: %.xml
    xsltproc -o $@ docbook-biblio-resources2docbook.xsl $<
```

Taper `make MONDOCUMENT-COMPLET.db` suffit alors à tout faire en cascade.

Comme tout le processus dépend de plusieurs fichiers, le script permet de créer les liens symboliques dans le répertoire courant, pointant vers le bon fichier.

1. Pour voir le RFC de numéro NNN, <https://www.ietf.org/rfc/rfcNNN.txt>, par exemple <https://www.ietf.org/rfc/rfc7749.txt>