

Using BibTeXML in DocBook XML to Write Scientific Articles



by Egon Willighagen
<egonw(at)linuxfocus.org>

About the author:

Got a masters degree in chemistry at the University of Nijmegen, and is doing his PhD research on molecular representation at the same University. Plays basketball and programs Java applications.



Abstract:

In this article I will show you how you can use a BibTeX like reference system with DocBook XML. To do this I have developed tools to ease this process, which are packages in the JReferences distribution.

Introduction

Latex users know how useful BibTeX is. It is a very convenient tool to add references to other scientific literature without caring much for the actual output, as it will be correct anyway - i.e. giving the correct settings, but *without* any manual type setting. Just like Latex itself. Moreover, in scientific literature it is common to number references with superscript numbers, like this¹. And these numbers should be in consecutive order. BibTeX takes care of this too.

DocBook is growing to become my favorite text authoring tool more and more everyday, because of the clean XML based syntax, the great support for making websites (e.g. the CDK website, <http://cdk.sf.net/>, is completely written in DocBook) and man pages. Next step for me was to use DocBook for writing scientific articles. Thus, I needed BibTeX for DocBook. Hence, I wrote JReferences.

JReferences does a bit more than BibTeX does. Like BibTeX it, it has tools to autonumber references from a plain text database, but it has more. It supports more formats (both input and output), it has a

MySQL backend which can be accessed with a PHP frontend. It tries to be a reference database too, like EndNote. However, while it is a open source project (GPL license) it has not yet attracted many developers other than myself, development is going slowly. This does not mean that it is not useful, though, and I will show you otherwise in this article.

When this article was published JReferences has been developed up to version 0.7.2. And this article considers that version.

A DocBook Article

Consider the example that can be found in the JReferences package.

```
<?xml version="1.0"?>
<!DOCTYPE article PUBLIC
"-//JReferences//DTD DocBook JReferences Module //EN"
    "../dtd/jreferences.dtd" []>
<article>

  <jref:mode>Number</jref:mode>

  <articleinfo>
    <title>Test Article</title>
    <author><firstname>Egon</firstname>
      <surname>Willighagen</surname></author>
    <date> 3 May 2000</date>
  </articleinfo>

  <section>
    <title>Some section</title>
    <para>This is a text with a reference
    <jref:cite id="Steinbeck99"/>.</para>
    <para>And now for some more serious tests, we
    add a second reference <jref:cite id="Bachrach99"/>.
    And again the first reference <jref:cite id="Steinbeck99"/>.
    </para>
  </section>

  <jref:reflist/>

</article>
```

I'll comment this example line by line. The first line is your common (optional) line denoting the syntax of this file, being XML. The second to fourth line state that the XML language that is used is DocBook, but that instead of the normal DTD the JReferences module is used. Normal DocBook XML does not know about JReferences, and using that DTD would give invalid documents. The JReferences module, however, knows about both DocBook as JReferences (for DocBook insiders: not about SVG or MathML yet). Hence, using that module gives you the possibility of validating your document again. The above example is a valid DocBook document.

The 5th line contains the starting tag of the article element. So far, so good. The 7th line, that is where the fun begins: the first jref element. The <jref:mode> element is used to tell JReferences which type of reference number markup it should use. In the introduction I already wrote that superscripted numbers

are normally used. There are many alternatives, though. JReferences supports [1], ¹, and [Steinbeck99]. The latter gives the code used in the reference. The example uses the first option.

The next few lines contains some basic DocBook content, and the next really interesting line is line 17. In this line the first reference is cited. Latex users would use `\cite{}` for this; the JReferences syntax is `<jref:cite id="SomeID"/>`. The ID corresponds to a reference in the database, which will be explained later. The next paragraph of the section contains another two citations, one being the first reference again.

To include the actual references, `<jref:reflist/>` is used in line 24. This JReferences command will convert to a DocBook formatted list of references, in the order in which they were cited.

The BibTeX database

The JReferences system needs a database, much like the *.bib files in Latex/BibTeX. JReferences has support for a BibTeX database backend, but for others too (like MySQL). BibTeX was developed by Vidar Gundersen and Zeger Hendrikse. The example in the JReferences distribution (0.7.2) does not use BibTeX yet, but the example article would have a BibTeX file like:

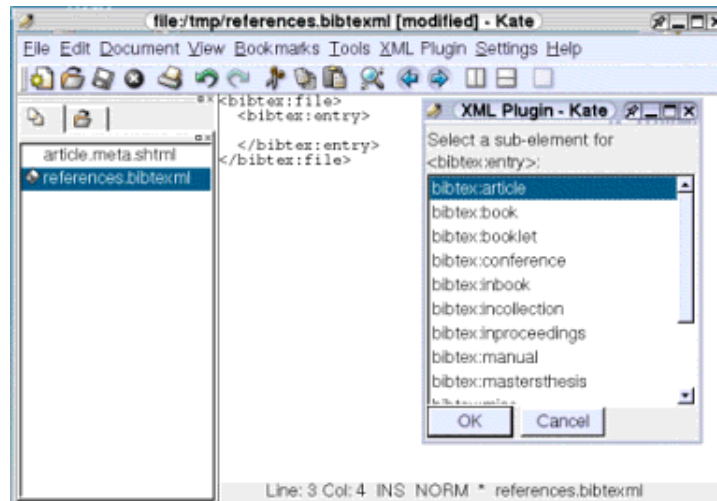
```
<?xml version="1.0" encoding="UTF-8"?>
<bibtex:file xmlns:bibtex="http://www.bitjungle.com/~bibtex/">

<bibtex:entry bibtex:id="Steinbeck99">
  <bibtex:article>
    <bibtex:title>JChemPaint - Using
      the Collaborative Forces of the Internet to
      Develop a Free Editor for 2D Chemical
      Structures</bibtex:title>
    <bibtex:author>Steinbeck, C. and
      Krause, S. and
      Willighagen, E.</bibtex:author>
    <bibtex:year>2000</bibtex:year>
    <bibtex:volume>5</bibtex:volume>
    <bibtex:pages>93-98</bibtex:pages>
  </bibtex:article>
</bibtex:entry>

<bibtex:entry bibtex:id="Bachrach99">
  <bibtex:article>
    <bibtex:title>End-User Customized Chemistry Journal
    Articles</bibtex:title>
    <bibtex:author>Bachrach, S. and
      Krassavine, A. and
      Burleigh, D.</bibtex:author>
    <bibtex:journal>J.Chem.Inf.Comput.Sci.</bibtex:journal>
    <bibtex:year>1999</bibtex:year>
    <bibtex:volume>39</bibtex:volume>
    <bibtex:pages>81-85</bibtex:pages>
  </bibtex:article>
</bibtex:entry>

</bibtex:file>
```

The second line contains the start tag for the root element `<bibtex:file>`. Such a file contains one or more `<bibtex:entry>` elements. And each entry consists of an BibTeXXML references type: article, book, inbook, incollection, unpublished, misc and others. Each such reference contains specific elements for that type, but a number of them are common, like `<bibtex:title>` and `<bibtex:year>`. The JReferences distribution includes the BibTeXXML DTD so that any DTD aware XML editor can easily edit BibTeXXML documents. Moreover, JReferences contains Meta DTD's for Kate in KDE 3.x (see Editing DocBook XML Documents) which are automatically installed in `$HOME/.kde/share/apps/katexmlplugin`.



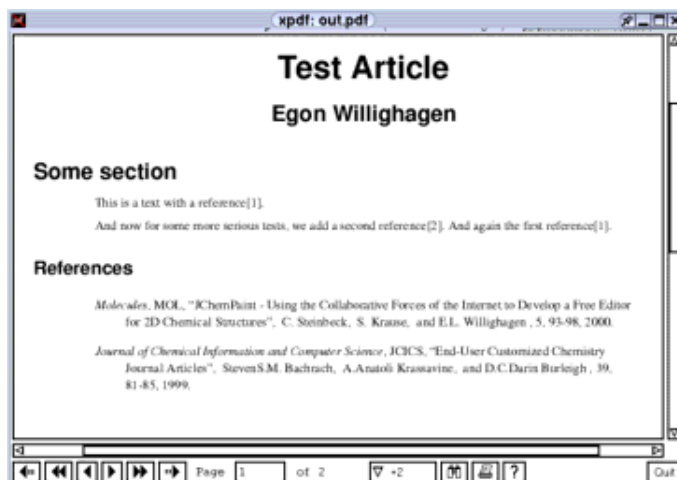
Editing BibTeXXML files with Kate, its XML plugin and JReferences' BibTeXXML Meta DTD.

Generating a Bibliography

Consider the two example files above. The DocBook document is saved as `article.docbookxml`, and the reference database is saved as `references.bibtexml`. JReferences does not yet contain a tool like the `bibtex` program, but the same can be achieved in a few command's. The below command's assume that you have installed JReferences on a Unix-like system, like Linux (see below):

```
jref-clear --filedb
jref-set --filedb --bibtexml references.bibtexml
jref-number --filedb article.docbookxml > article-numbered.docbookxml
```

The resulting file called `article-numbered.docbookxml` is a valid DocBook XML 4.1.2 document without any `<jref:*>` element and can be processed by any other tool used to convert DocBook XML documents to, for example, PDF. (See for example Making PDF documents with DocBook).



The resulting PDF with numbered references and an included bibliography.

That is all you need to know. Or, actually...

Formating Styles

There is one more interesting thing. BibTeX supports styles, because most journals have specific requirements on how the bibliography is formatted. JReferences contains only two styles at this moment. The first is some default DocBook XML format, which is not really a style. But there is also a style required by the American Chemical Society (ACS) which is also available in JReferences.

The `<jref:reflist>` element has a `@style` attribute with which you can set the style to use instead of the default style. To use the ACS style you need to replace line 23 with

```
<jref:reflist style="ACS"/>
```

Installing JReferences

JReferences requires a Java 1.3 (or higher) installation, Xerces, Log4J and the DocBook XML DTD 4.1.2. Some tools require additional tools, like python (for BibTeX 2 BibTeXXML conversion), and Perl (for cleaning up EndNote's BibTeX output).

If those are installed, JReferences can be installed by doing:

```
./configure --prefix=$HOME  
make  
make install
```

If some tools are not found, try these options: `--with-xercesdir`, `--with-log4javadocir` and `--with-sgmlmdir`. For more information about these options, type `./configure --help`.

The Project

JReferences is now about two year old, and though it has been downloaded many times I do not get much feedback. That is, except for my personal experiences. In the last few months JReferences has been successfully used for writing a real scientific article. However, like any good open source project, any comment, bug reports, patches, ideas and success stories are welcomed at the JReferences Project Site.

References

- DocBook XML, <http://docbook.sf.net>
- BibTeX XML, <http://bibtexml.sf.net>
- JReferences, <http://sf.net/projects/jreferences>

<p>Webpages maintained by the LinuxFocus Editor team © Egon Willighagen "some rights reserved" see linuxfocus.org/license/ http://www.LinuxFocus.org</p>	<p>Translation information: en --> -- : Egon Willighagen <egonw(at)linuxfocus.org></p>
---	--