

• Snowbooks' submission for the innovation in the book industry award: XSL Project

The judging requirements for this award require us to demonstrate 'outstanding creativity in digital innovation'. Our submission demonstrates not only creativity, but pragmatism, efficiency, and cost-saving. An aspect of Snowbooks on which we pride ourselves is that we do not have a separate digital strategy, distinct from business as usual. Instead, computers and programs are the tools we use to make our business as efficient and interesting as possible; to free up our time to focus on the more creative side of the business that gives us our spark and to generate widespread interest.

The innovation that we will describe here has saved us, conservatively, a month a year. It has got us extra sales from home and overseas, and has provided us with an invaluable selling tool for retailers, wholesalers, libraries, non-book sellers, other merchants and individuals. It has garnered us column inches and speaking engagements. It has avoided the untold cost of lost sales through inaccuracy or missing data. It has require us to think outside of the box, to suspend disbelief in our own abilities and to make tangible an ambitious dream.

We have developed the capability - without using external consultants, just by reading books - to produce all our paper and online book marketing materials (webpages, AIs and catalogues, in .html and .pdf form) in a four step process that takes two minutes. The process is as follows and the total number of mouse clicks involved at each stage is given in brackets:

First, input all bibliographic data into the database. We use the Anko Publishing Manager database.

- » Export all title's records from the database in XML format (2)
- » Open the batch file to run the XSL (2)

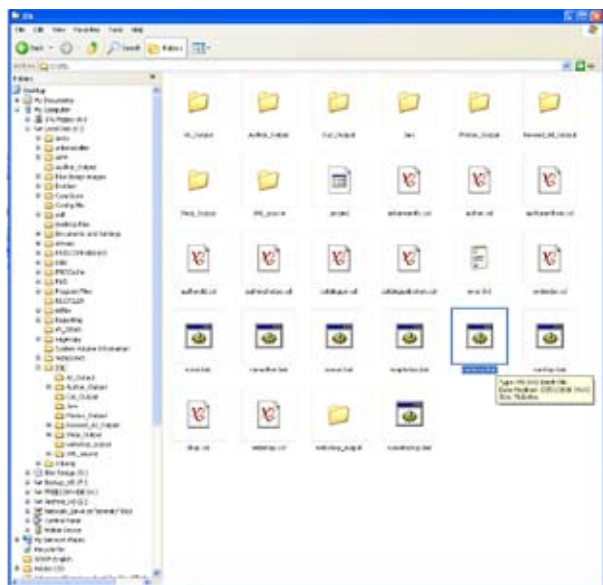
The output file can then be uploaded to snowbooks.com to form part of the catalogue. In total, it requires 4 clicks, which is very swift. (Needless to say, we also use the original XML (ONIX) file to populate our Nielsen records.)

XML is short for Extensible Mark Up Language. It's the language that ONIX is written in, and is a way of tagging up content so that you know what the content is. For instance, here is a way to describe a book:

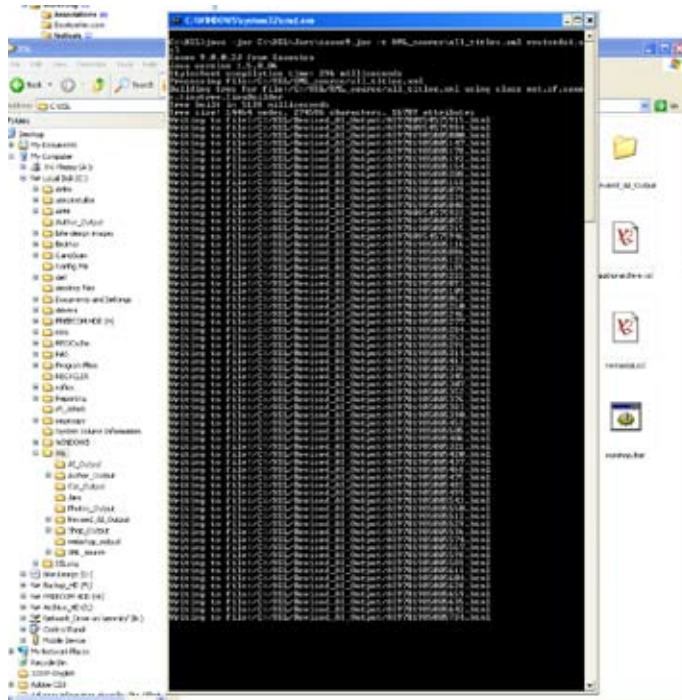
```
<title>Persuasion</title>
<author_first_name>Jane</author_first_name>
<author_last_name>Austen</author_last_name>
<ISBN>9781905005111</ISBN>
```

It's not very complicated, just some content wrapped in tags.

XSL, in turn, understands the XML tags and does what it's told with them. For instance, if we want a block of important bibliographic data to feature on our end document, we get the XSL to look for the tags that show the content is, say, ISBN, Pub date, Format, Size and Number of pages, and to display the data from within the tags.

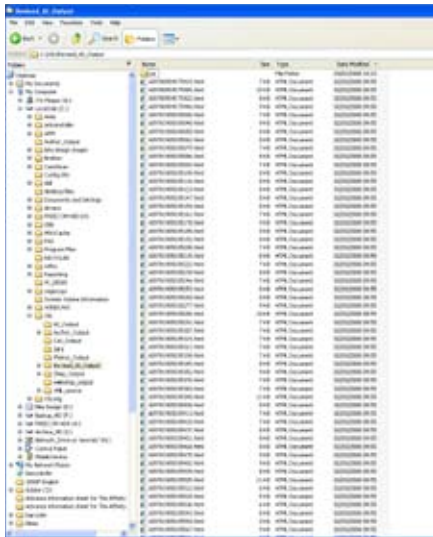


Double click on the batch file, which runs the XSL



The batch file running, creating one file per title. It took me four goes to capture this screen shot as it runs so quickly!

The newly created AIs, sitting in their folder. Double click on one to open and you see...



...The result: an html file (webpage) with data drawn from the original database. Pictures and formatting defined by CSS (cascading style sheets, the gold standard for web design). This is an AI, but we use the same process for author pages, web pages, online shop pages, catalogue pages and photo pages.



What are the main benefits?

- » It's very quick.
- » It's very accurate. No re-keying of data means no mistakes - only the original database needs to be correct, not ten versions dotted around in different files
- » This accuracy means that we aren't losing sales from wrong or absent data. Lost sales are hard to put a value on, but it's an important point.
- » It's consistent. Each page has the same type of data.
- » It makes our information available for anyone with a web browser to see. I can email a URL of an AI to a customer and they have it immediately. That has been particularly helpful in overseas sales and sales to non-book retailers.
- » Once the data is in an XML format, it can be used to populate other files. For instance, we also use XML to fill in Indesign catalogue layouts in a similarly rapid process, and the many different spreadsheets that the different retailers require for titles to be considered for selection. We also use it to populate Word templates - see below.
- » It has got us recognised as leading the market - not just among independent publishers, either. I was invited to talk about this at the E4Books seminar in the summer and the feedback was inspiring. Some professional IT managers didn't know this stuff so straightforward or productive.

How did we do it? We had bought the Anko Publishing Manager so had a database containing all our data. It seemed daft to have to keep typing out data for things like AIs, catalogues and so on, so we started to learn about this thing called XSL. We bought a book or two on the subject, read them, and then wrote the code. It's interesting: I am an arts graduate, running a book publishing company, with a love of literature. I am in no way a geek or a technically-minded person, and have no IT training. But with a bit of patience and a burning desire to make my company as efficient as possible, I have learned how to write code from scratch. The motivation was threefold:

- » **I hate wasting time.** I'd much rather spend a weekend learning how to use XSL, or another way of automating something, than spend one day a week, every week, retyping or copying and pasting data from one document to another
- » **I hate making mistakes.** Rekeying data, no matter how careful you are, simply always results in one or two typos. Before you know it, the data on Nielsen, Amazon, Waterstone's.com and your own website is subtly different. And what happens when the publication date changes, or the page extent? Rather than trying to correct every bit of paper, every partner's website and records, it's much more simple and reliable to have one version of the truth.
- » **I hate spending money.** I would much rather read a big thick book on XSL than pay a consultant £3000 to write an application for me. Firstly, we'd spend £3000 we didn't have to. Secondly, we wouldn't have the knowledge to write the next useful application that I want. By reading the books, and learning this ourselves, we're adding to the knowledge assets of Snowbooks and increasing our value as a business.



XML used to populate fields in a Word document

I thought long and hard about which initiative to submit for this award: we have done lots of innovative things, from creating the first Second Life publisher's bookstore to launching the SnowCase, a new way for unpublished authors to get their work showcased on our blog. But I think that using XSLT to generate all our marketing materials has saved us more money and time than any other initiative we have done, and is a great example of how you don't need formal IT training, or a budget of any description - just effort and determination - to have world-class data management.