

Extreme automation for requirements documentation

Scriptorium completely automated a client's process for publishing requirements documents. The new 15-minute process replaces labor-intensive tasks that took a week to complete.

Our client develops large software applications for military use. During the planning process, software requirements are stored in a specialized database, which can output information into Microsoft Word documents.

Requirements engineers typically cut and paste input for the database from Word and other sources, which results in formatting inconsistencies. Output generated from the database required significant manual reformatting to remove these inconsistencies. In addition, our client identified inefficiencies with manually added tables and graphics and unresolved cross-references. Correcting all of these issues required up to a week of work.

While developing requirements for the software, teams meet daily to create and capture new information. Our client requested that we reduce the document generation time so that updated output could be distributed daily.

Our client's wish list included the following:

- ❖ Produce an Eclipse plug-in with consistent formatting from the default database output (Microsoft Word files).
- ❖ Deliver documents overnight.
- ❖ Make use of an existing Darwin Information Typing Architecture (DITA) Open Toolkit implementation.
- ❖ Use only the client's existing software licenses, open source software, or scripts written by Scriptorium—no additional software purchases.

Solution strategy

Our solution was constrained because we had to use Word output from the requirements database as a starting point. We needed to migrate these documents to structured DITA topics with a topic map. The resulting Extensible Markup Language (XML) files could then be processed through the client's existing process based on the DITA Open Toolkit. We broke down the solution into several steps:

- ❖ Acquire the graphics from the Word document for subsequent reuse in DITA XML.
- ❖ Convert the Word document to a form of XML.
- ❖ Convert the XML to DITA.
- ❖ Link the graphics into the DITA XML file.
- ❖ Break up the single DITA document into separate topic files and create a topic map to organize them.
- ❖ Feed the output into the existing DITA Open Toolkit implementation for formatting.
- ❖ Provide a framework for controlling the flow of processing through these steps.

Implementation

Because the DITA Open Toolkit already uses Ant for automating build processes, we chose it to control the processing flow. Issuing a single Ant command starts the conversion process for a document. Additional tools included the following:

Microsoft Word macro: The macro saves the Word document to HTML and extracts individual graphic files for any embedded graphics. The process also creates a list of graphic files that could be used for the subsequent relinking process.



Open Office Writer macro: The macro saves the document to DocBook XML. Opening Word files in Open Office and then saving to DocBook XML is a reliable method for extracting XML content from Word files. In particular, this approach yields good results with tables.

DocBook-to-DITA plug-in: This prototype DITA Open Toolkit plug-in converts DocBook output to DITA. Only minor tweaking of the DocBook-to-DITA transformation files was needed.

Custom Extensible Stylesheet Language (XSL) transformations: We wrote custom XSL transformations to restructure content and relink graphics into the XML files.

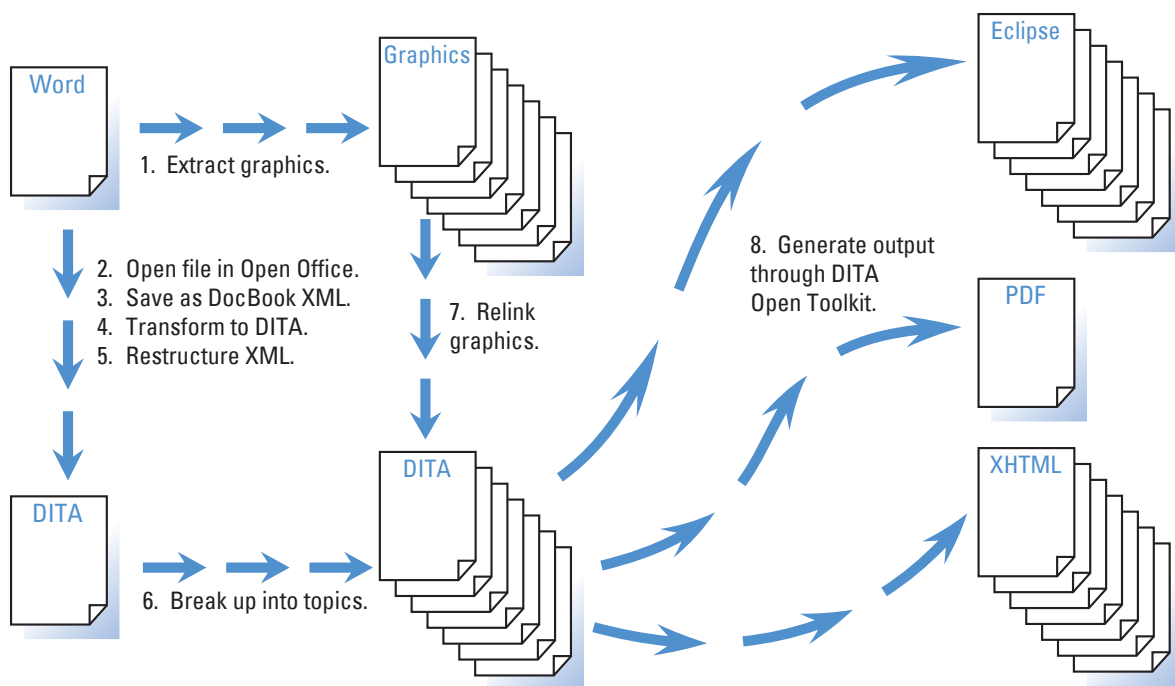
Custom Perl script: We wrote a custom Perl script to produce individual topic files from our large DITA files and create the corresponding DITA topic map files. It would have been possible to break up the content through XSL, but we judged Perl to be more efficient and a better match for this particular

problem. (Generally, Perl is better at file-processing tasks than XSL.)

Results

Our client began using a prototype system to support a new round of requirements development reviews. After each day's updates are entered into the requirements database, four documents are generated from the requirements database and processed as shown in the figure. Processing for each document requires approximately 5 to 15 minutes. The Eclipse plug-in for each document is posted, and participants can see changes from the previous day's reviews to prepare for the next day's work.

Our client is investigating how to extend this process to streamline other Word-based documentation efforts. They are also looking into producing DITA XML directly out of the requirements database. We have continued to refine this process; for example, we eliminated the Word macro and now extract the graphics with an Open Office macro.



scriptorium.com

