An OASIS DITA Adoption Technical Committee Publication

DITA 1.3 Feature Article:
A Brief Introduction to XSL for Processing DITA Content

Author: Leigh W. White
On behalf of the DITA Adoption Technical Committee

Date: 28 March 2017

OASIS (Organization for the Advancement of Structured Information Standards) is a not-for-profit, international consortium that drives the development, convergence, and adoption of e-business standards. Members themselves set the OASIS technical agenda, using a lightweight, open process expressly designed to promote industry consensus and unite disparate efforts. The consortium produces open standards for Web services, security, e-business, and standardization efforts in the public sector and for application-specific markets. OASIS was founded in 1993. More information can be found on the OASIS website at http://www.oasis-open.org.

The OASIS DITA Adoption Technical Committee members collaborate to provide expertise and resources to educate the marketplace on the value of the DITA OASIS standard. By raising awareness of the benefits offered by DITA, the DITA Adoption Technical Committee expects the demand for, and availability of, DITA conforming products and services to increase, resulting in a greater choice of tools and platforms and an expanded DITA community of users, suppliers, and consultants.

DISCLAIMER: All examples presented in this article were produced using one or more tools chosen at the author's discretion and in no way reflect endorsement of the tools by the OASIS DITA Adoption Technical Committee.

This white paper was produced and approved by the OASIS DITA Adoption Technical Committee as a Committee Draft. It has not been reviewed and/or approved by the OASIS membership at-large.

Document History

Revision	Date	Author	Summary
First Draft	28 Mar 2017	Leigh W. White	Initial draft
Second Draft	30 Mar 2017	Leigh W. White	Incorporating edits based on comments from Nancy Harrison
Third Draft	9 Apr 2017	Leigh W. White	Incorporated edits based on comments from Ian Balanza.

Revision	Date	Author	Summary
Third Draft	12 Feb 2018	Leigh W. White	Added OASIS frontmatter amd minor tweaks.
Fourth Draft	16 Feb 2018	Leigh W. White	Incorporated edits based on comments from Tom Magliery.
Fifth Draft	12 Apr 2018	Leigh W. White	Incorporated edits based on comments from Eliot Kimber.

Table of Contents

What is XSL(T)?

If you have ever used the DITA Open Toolkit to transform DITA content into HTML, online help, PDF, or any other output type, you have used XSL. If everything went well, you probably didn't think too much about how the transformation happened. If things did not go well, you probably gave the transformation a little more (unkind) thought.

If things did not go well, were you inclined to pass the problem off to someone else on your team? Or did you want to have a look under the hood to examine the DITA Open Toolkit (DITA OT) to understand the transformation? The DITA OT is a collection of Java files, scripts, build files, and some other stuff, all of which prepare DITA files for transformation using XSLT.

This article first focuses on the basics of understanding XSL—the heart of the DITA OT—to transform your XML content, and then turns to a couple of examples of XSL within the DITA OT, specifically for transforming DITA content.

The simplest way to describe XSL is that it transforms DITA, or any other kind of XML, into something else—like HTML, Web help or a PDF. XSL has several parts:[1]

- XSLT, or **XSL T**ransformations. XSLT is a language for transforming XML documents.
- XPath, or the **XML Path** Language. XPath is an expression language used by XSLT to access or refer to parts of an XML document.
- XSL-FO, or **XSL F**ormatting **O**bjects. XSL-FO is an XML vocabulary for specifying formatting semantics. It's used as the basis for PDF transformation.

Informally, "XSL" often refers to XSLT only. In turn, "XSLT" often refers informally to both XSLT itself and XPath, since most XSLT documents use XPath expressions. XSLT is the focus of this article, which looks first at the basics of understanding XSL in general and then discusses XSLT within the DITA OT.

[1] These definitions are from the World Wide Web Consortium (W3C) (*https://www.w3.org*), which is an international community that develops open standards to ensure the long-term growth of the Web. Much like OASIS (Organization for the Advancement of Structured Information Standards, *https://www.oasis-open.org/*) develops and maintains the specifications for DITA, the W3C develops and maintains the specifications for HTML (Hypertext Markup Language), CSS (Cascading Style Sheets), XSLT, and other Web-related standards. Like DITA, the XSL technologies are open standards, and so their workings are transparent, platform independent, and have a robust architecture.

How does XSL turn your XML content into output?

Before talking specifically about the stylesheets in the DITA OT, which are a bit unusual in how they identify elements, it is important to have a general understanding of how XSL works. The first thing to understand is that XSL doesn't turn XML into something else by itself. A stylesheet is simply a static file. However, the stylesheet gives instructions to an XSLT processor such as Xalan (*https://xalan.apache.org/*) or Saxon (*http://saxon.sourceforge.net/*) and the processor uses those instructions to turn XML into HTML (or into XSL-FO for PDF output).

Note: Not all XSLT processors are created equal. Some work only with older versions of XSLT. Be sure to use a processor that's appropriate for the version of XSLT you're using.

If you have ever used a desktop publishing application, you should be familiar with templates. In the desktop publishing world, a template usually includes page definitions, paragraph and character formats, table formats, and so forth. As you edit your document, you apply styles manually based on your knowledge of how the text should look.

When you transform XML, you do not have the opportunity to apply specific styles to the text manually; you are depending on an automatic process to do that. You have to supply the process with the correct instructions, such as "When you encounter a title element, check to see if it is part of a topic that is nested in another topic. If so, indent it and make it bold and 14 points. If not, don't indent it and make it bold and 18 points."

These kinds of instructions are found in XSL stylesheets. Each stylesheet contains a number of templates—not to be confused with DTP templates!—most of which are designed to process a specific element. The templates

1. test an element for its context (because some elements should be processed differently depending on where they occur)
2. transform the DITA element into an HTML element or an XSL-FO element
3. apply formatting to the new element

XSLT processors iterate through the input XML and as they encounter a specific element, they find the matching template and follow the instructions in that template to create a corresponding HTML element with appropriate formatting attributes (or better, formatted by an accompanying CSS) or an XSL-FO element with formatting attributes that are appropriate for PDF output.

(PDF can also now be done with HTML and CSS, and that may well be the future path. For now, though, XSL-FO is still the more common approach for PDF output.)

This is a very general explanation. If you're curious to know more, the DITA OT documentation fully explains the whole process. This article is more concerned with explaining how XSLT itself transforms XML content to an output format, so let's move on to that.

The DITA OT stylesheets

Now that you've seen a basic XSL stylesheet, let's look specifically at stylesheets in the DITA OT.

When you process, or build output for a ditamap, you send it to the DITA OT. The DITA OT consists mainly of a bunch of build files and stylesheets. (It contains other important files as well, but they don't matter for this article.) These build files and stylesheets are grouped together into plugins. Generally, each plugin corresponds to a particular output type. When you specify that you want XHTML/HTML output, or Webhelp, or a PDF, or any other kind of output, you (or the tool you are using) sends an instruction to the DITA OT telling it which plugin to use in order to process the content you are sending to it. The specified plugin could be one of the default plugins that come with the DITA OT, or it could be a custom plugin developed specifically for your outputs.

Before we go into a discussion of the stylesheets, you should understand that there are a lot of things that happen when you first start a DITA OT build. Your content is validated, your map is parsed (and filtered, if applicable), keys and conrefs are resolved, and so forth. All of that is beyond the scope of this article. Here, we're going to focus just on how the stylesheets in the applicable plugin turn content from DITA XML to XHTML or to a PDF.

To keep things simple for this article, we'll look at only two of the DITA OT plugins: the one that generates XHTML (**org.dita.xhtml**) and the one that generates PDF (**org.dita.pdf2**).

If you're creating XHTML-based output, the DITA OT outputs one XHTML file per DITA topic (unless you specify otherwise). It's sufficient to transform the DITA XML directly to XHTML and that's mostly what the templates in the **org.dita.xhtml** plugin do.

If you're creating PDF output, the process is a little more complex. You don't (usually) want a separate PDF file for each topic; you want one PDF for the entire map. The first step, then, is to merge the entire map into one big DITA XML file. That merged file is then sent forward for processing. To create a PDF from DITA XML, you need a separate application called a PDF renderer. PDF renderers do not understand DITA XML, so the next step is to transform the DITA XML into another form of XML called XSL-FO. This is what most of the templates in the **org.dita.pdf2** plugin do. The resulting XSL-FO file is then sent to the PDF renderer and it creates a PDF.

You might guess at this point that the templates in **org.dita.xhtml** and **org.dita.pdf2** are going to be rather different because they are doing different kinds of transforms—DITA to XHTML and DITA to XSL-FO. You'd be correct, although they all have one important thing in common: they all use XSLT and XPath.

The simple addressbook example we looked at consisted of a single stylesheet. You'll quickly notice that each of the plugins in the DITA OT consists of multiple stylesheets because it would be far too impractical to include hundreds of templates in a single stylesheet. A stylesheet can import or include other stylesheets so that they all end up functioning as if they were a single stylesheet.

What to do next

If you're not responsible for maintaining the stylesheets for your content output, you can simply relax and enjoy this new bit of high-level XSL knowledge. However, if you're curious to know more, you can continue to research XSL on the Web or you can roll up your sleeves and dig in a little bit. There are many, many online and print resources available for learning XML and XSL...and many of them are free!

Start simple. Create a simple XML file of your own (perhaps your own addressbook, or list of DVDs, or international capitals with their populations and elevations, or whatever interests you). Then create a basic stylesheet to transform them to HTML. (Sometimes it can be helpful to start by creating the HTML file to get the output structure correct and then work it into XSL templates.)

If you are using an XML editor that includes transformation capability then you don't have any extra setup to do for your practice. One or more XSLT processors and all the necessary associated files are already installed and ready to go.

After you've had a little hands-on experience, you can dive into working with DITA OT stylesheets. If you have a custom plugin, ask its owner if they can create a sandbox copy for you to play around with. Or create your own! The DITA OT documentation (in the `[DITA-OT]\doc\dev_ref` folder) explains how the DITA OT plugins work and how to create your own. Specifically, take a look at the plugins-overview.html topic. For a deep dive into PDF plugins, *DITA For Print: A DITA Open Toolkit Workbook*, by Leigh White (yours truly), available from *XML Press*, offers a full explanation of how the **org.dita.pdf2** plugin works and provides step-by-step instructions for creating your own PDF plugin.

Good luck!