Creating PDF files from DITA content

WHITE PAPER

Sheila Loring
Senior Technical Consultant

David James Kelly
Project Manager/ Technical Consultant with

Sarah O'Keefe
President

Alan Pringle
Director of Publishing Operations

When technical documentation groups adopt structured authoring, they often choose the Darwin Information Typing Architecture (DITA) open-source standard because it requires no up-front investment. The DITA Open Toolkit (DITA OT) provides a way to produce multiple outputs, including Portable Document Format (PDF) files; however, the technology for creating PDF files is limited, and modifying the formatting is challenging.

Software companies have stepped in to offer methods for customizing and producing PDF files, but there are different levels of DITA support and various degrees of integration with the DITA OT.

How do you decide which method is best for you? This paper explains the alternatives and trade-offs for each method and helps demystify the decision process.

Typography versus velocity

Documentation is an important product in many industries—including finance, software, and telecommunications. Readers of the documentation often expect professional, precise page design. Mixed column layouts, uncontinuous text flows, and controlled kerning are hallmarks of this type of publication. These and other factors push some publishers to higher levels of quality, variety, and flexibility in their layout designs.

At the same time, the volume and velocity of publication can be overwhelming. Particularly in technical communication, manuals consisting of thousands of pages might be republished as often as every night. Customization of documents for multiple customers requires simultaneous publication of many content variants. It is not unusual for the instructions for an inexpensive product, such as a bicycle headlight, to be printed in 20 languages. Not long ago, these kinds of publication

feats would have been considered unthinkable or would have required large departments of production editors. Those departments have by and large disappeared, but the requirements are still there and have, in fact, increased.

The following factors must be considered when choosing a path from DITA to a PDF file:

- Some publications need complex layouts. They have articles linked from page 1 to page 5, images floating in text, sidebars, pull quotes, unique layouts on each page, and precisely spaced ligatures. Popular magazines are on the high end of these complexities.
- Some publications need velocity, volume, and versioning. Velocity is the frequency at which content is published, volume is the amount of content published, and versioning refers to content variants such as language and audience. Technical documentation has these requirements.

The choices for creating PDF files from DITA are limited by the relative youth of DITA as an architectural standard (introduced by IBM in 2001 and approved as an OASIS standard in 2005). The choices are also due in part to the complexity of PDF file conversion. PDF files require much more sophisticated layout instructions than HTML.

The basic paths for getting from DITA to a PDF file are:

- The DITA OT with an Extensible Stylesheet Formatting Objects (XSL-FO) processor (Apache FOP, Antenna House XSL Formatter, and RenderX XEP).
- XML authoring tools that work with the DITA OT and an XSL-FO processor to produce PDF files. They let you author XML and create PDF files from an interface (Just Systems XMetaL Author, SyncRO Soft oXygen, and Quark DITA Studio).
- Conversion tools that produce PDF files from DITA files (WebWorks ePublisher).
- Help authoring tools that import DITA files and have built-in PDF file conversion capabilities (MadCap Flare and Adobe RoboHelp).
- Page-based publishing software that imports DITA files and has built-in PDF file conversion capabilities (Adobe InDesign and FrameMaker).
- High-capacity XML publishing tools that work with the DITA OT and produce PDF files (SDL XySoft XML Professional Publisher and Arbortext Publishing Engine).

Each path has its own advantages and disadvantages. For example, FOP is free, but it is more limited than commercial tools. InDesign produces a beautiful PDF file; however, importing DITA involves extensive scripting. The path you choose depends largely on what you're willing to compromise (for example, automation at the expense of perfectly designed PDF file).

Another factor in choosing a conversion method is overall cost. Some solutions, such as the DITA OT, are initially free, but modifying them is difficult and expensive. Other solutions cost more, but you may be happier with their performance out of the box, or they may provide other functions you need.

The following table shows the list price of all tools covered in this paper. Tools are listed alphabetically.

Product	List price
Arbortext Publishing Engine	Not available*
DITA Studio	$2500
ePublisher	$85 monthly for each user
Flare	$899
FOP	Free
FrameMaker	$999
InDesign	$699
oXygen	$64–$449
RoboHelp	$999
XEP	Free–$4400
XMetal Author	$1195
XML Professional Publisher	Not available*
XSL Formatter	$1250–$5000

* The vendor didn't provide pricing.

So, how do the different types of solutions measure up?

Support of DITA features

Many DITA-capable tools support DITA features out of the box. For example, most of them support DITA maps and conditional processing with no customizations to the programs. Other tools, such as InDesign, require scripting in order to process maps, conrefs, conditions, and specializations.

The DITA OT natively supports DITA maps, conrefs, conditional processing, and specialization. The following table shows which major DITA features are supported in the software programs reviewed in this paper:

	DITA Maps	Conrefs	Conditional Processing	Specialization
XMetaL Author	X	X	X	X
oXygen	X	X	X	X
DITA Studio	X	X	X	X
ePublisher	X	X	X	X
RoboHelp	X	X	X	X
Flare	X	X	X	

	DITA Maps	Conrefs	Conditional Processing	Specialization
InDesign				
FrameMaker	X	X	X	X

The DITA OT and an XSL-FO processor

The DITA OT with an XSL-FO processor is a commonly considered solution. The DITA OT comes with XSL-FO transforms, which are run on the DITA XML to produce an FO file. The FO file contains the DITA content and instructions about how the content should be formatted. The XSL-FO processor then converts the FO to a PDF file.

All PDF file formatting is controlled by the DITA OT XSL-FO. You edit the XSL-FO in an XML editor to modify the formatting. For example, Figure 1 shows the FO for calculating table column widths. Modifying this code might seem daunting for the average user.

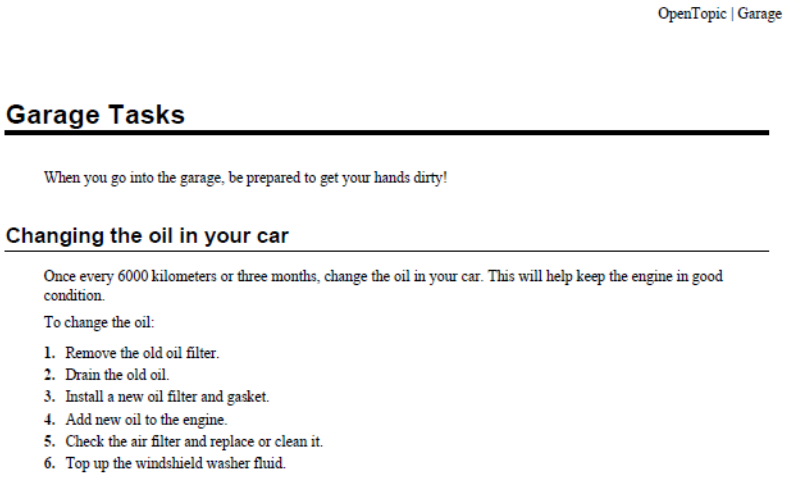
Figure 1: Example of XSL-FO in the DITA OT

```
<xsl:template name="calc.column.width">
 <xsl:param name="colwidth">1*</xsl:param>
 <!-- Ok, the colwidth could have any one of the following forms: -->
 <!--       1*      = proportional width -->
 <!--     1unit     = 1.0 units wide -->
 <!--       1       = 1pt wide -->
 <!--  1*+1unit     = proportional width + some fixed width -->
 <!--     1*+1      = proportional width + some fixed width -->
 <!-- If it has a proportional width, translate it to XSL -->
 <xsl:if test="contains($colwidth, '*')">
  <!-- modified to handle "*" as input -->
  <xsl:variable name="colfactor">
   <xsl:value-of select="substring-before($colwidth, '*')"/>
  </xsl:variable>
  <xsl:text>proportional-column-width(</xsl:text>
  <xsl:choose>
   <xsl:when test="not($colfactor = '')">
    <xsl:value-of select="$colfactor"/>
   </xsl:when>
   <xsl:otherwise>1</xsl:otherwise>
  </xsl:choose>
  <xsl:text>)</xsl:text>
 </xsl:if>
 <!-- Now get the non-proportional part of the specification -->
 <xsl:variable name="width-units">
  <xsl:choose>
   <xsl:when test="contains($colwidth, '*')">
    <xsl:value-of select="normalize-space(substring-after($colwidth, '*'))"/>
   </xsl:when>
   <xsl:otherwise>
    <xsl:value-of select="normalize-space($colwidth)"/>
   </xsl:otherwise>
  </xsl:choose>
 </xsl:variable>
```

With respect to page layout sophistication, automation, and cost, the DITA OT and XSL-FO option provides the following:

- Generic but moderately customizable output. The XSL-FO standard is purposely limited in certain areas of page formatting.[1] Figure 2 shows the default formatting for a PDF file produced with FOP from the DITA OT.
- High levels of automation. The DITA OT comes with Ant scripts that automate the conversion process. You run the Ant scripts, and Ant calls the DITA OT Extensible Stylesheet Transformations (XSLT) and FO processor to produce PDF output.
- Free or inexpensive initial entry is possible, though most commercial processors require some investment and can cost up to $5000. XSL-FO stylesheets are also available from commercial providers, some of whom provide the stylesheets for free. Low cost is based on minimal customizations.When extensive stylesheet modifications are required, costs increase.

Figure 2: *A PDF file produced in the DITA OT 1.5 using FOP*


OpenTopic | Garage Tasks | 3

Garage Tasks

When you go into the garage, be prepared to get your hands dirty!

Changing the oil in your car

Once every 6000 kilometers or three months, change the oil in your car. This will help keep the engine in good condition.

To change the oil:

1. Remove the old oil filter.
2. Drain the old oil.
3. Install a new oil filter and gasket.
4. Add new oil to the engine.
5. Check the air filter and replace or clean it.
6. Top up the windshield washer fluid.

If the plain PDF file shown in Figure 2 doesn't meet your needs, you'll modify the XSL-FO to customize components such as headers and footers, logos, and attractive fonts.

Because of the potentially low initial cost and the high levels of automation, the XSL-FO solution is popular with technical publishers. However, if you are accustomed to tweaking page layout and creating highly designed pages, expect to lower your standards.

FOP

FOP is an open-source Java-based processor available for free from the Apache XML Project (http://xml.apache.org/fop/). FOP can produce usable output, but it is still under development and has some limitations. For example, the XSL-FO 1.1 specification is supported, but FOP doesn't support automatic table layout and floating images. Nevertheless, many users choose FOP because it is free. PDF files can be produced from the command line, or the library can be used in servlets and other Java applications. It runs on Windows, Solaris, Mac, and UNIX.

1. Refer to "http://en.wikipedia.org/wiki/XSL_Formatting_Objects#Drawbacks_of_XSL-FO." For example, XSL-FO doesn't support "page-aware" layout, such as making a figure appear on the page opposite its anchor. It also cannot create areas with different numbers of text columns on a single page, and it cannot wrap text around both sides of a floating object.

XSL Formatter

XSL Formatter, which has comprehensive FO compliance, is from Antenna House, Inc., (http://www.antennahouse.com). The company provides a large number of XSL-FO extensions, such as alternate text, annotations, form fields, footnote autonumbering, and revision bars. They also provide a set of DITA stylesheets that could be used as a starting point as an alternative to the DITA Open Toolkit. The DITA stylesheets can be downloaded for free.

The software runs from the command line or an interface on Windows and Windows 64-bit. On other supported platforms (Solaris, Linux, Linux 64-bit, Macintosh, HP-UX and AIX), the program runs from the command line or from another program.

XEP

XEP is a commercial product from RenderX (http://www.renderx.com). It is a Java-based processor compatible with most platforms and can be run from an interface or the command line. XEP is regularly updated to improve standards conformance and to comply with strict requirements in PDF file formatting. XEP conforms to the XSL-FO 1.0 specification. Indexing and bar code extensions, among other FO features, are supported.

DITA-capable XML authoring tools with PDF file conversion

This solution includes the DITA OT and XSL-FO processor with an XML authoring environment. The DITA OT is distributed with these programs, and the programs use the OT to produce output.

Formatting is completely controlled through the DITA OT XSL-FO. The default formatting is rather generic, as shown in Figure 2 on page 5.

Three tools in this category include XMetaL Author, oXygen, and DITA Studio. These solutions have the same formatting and automation characteristics as the DITA OT with FO processor solution. See “The DITA OT and an XSL-FO processor” on page 4 for details.

The following characteristics are common among XML authoring tools:

- The cost ranges from $64 to $2500.
- Authors can edit the XML code directly or in a view that displays the elements (tagged view).
- The programs come with XSL-FO processors and produce PDF files within the interface. XMetaL ships with XEP, and oXygen comes with FOP. DITA Studio uses FOP.
- Because the XML authoring tools are developed to work with a specific version of the DITA OT, it might be difficult to upgrade the OT without upgrading the tool.
- These tools have a word processor-like view, for example, bold and italic buttons and drop-down lists for applying styles.
- All programs in this category can be run from the command line to automate PDF production.

Consider these solutions if you are looking for an integrated XML authoring and production environment.

XMetaL Author 5.5

XMetal Author is an XML authoring tool integrated with the DITA OT. It transforms XML into an FO file and then uses XEP to convert the FO to a PDF file. XMetaL Author also ships with FOP, and you can also configure XMetaL Author to use an external XSL-FO engine, such as XSL Formatter.

Because XMetaL Author is integrated directly with the DITA OT, you modify the FO to customize the PDF. See "The DITA OT and an XSL-FO processor" on page 4 for details.

XMetaL lets you modify the copy of the DITA OT that is installed with XMetaL, though we recommend you configure XMetaL to use your own customized or specialized version of the DITA OT.

By default, a PDF file generated from XMetaL Author includes bookmarks, thumbnails, a cover page, and a table of contents. The map title is also inserted in the PDF file footer.

In addition to handling specializations, XMetaL Author supports DITA maps, conrefs, and conditional processing out of the box.

There are a few options for specifying which conditional content to publish. One way is to select the conditions to show and hide. XMetaL Author then generates the ditaval file for you. You also have the option of exporting the conditions to a ditaval file, which is useful if you want to set advanced options in the ditaval file. On the other hand, if you know enough to customize the ditaval, you might prefer to select a ditaval you've already created.

Modifying the PDF file formatting via XSL-FO is no small task for most casual authors. As with other DITA OT-based authoring tools (and using just the OT), a programmer or writer trained in XSLT will need to make all customizations for XMetaL projects and also develop the specialization configuration files.

oXygen 9.3

oXygen is another XML authoring tool with integrated DITA OT functionality. DITA files are transformed into an FO file and then run through Apache FOP to convert the FO to a PDF file. You can also configure oXygen to use an external XSL-FO engine such as XEP. In fact, RenderX sells a package called Docbench that includes XEP processing with a license of oXygen.

A PDF file generated from oXygen includes bookmarks, thumbnails, a cover page, and a table of contents. Because oXygen is integrated with the DITA OT, you modify the XSL-FO to include other components in the PDF file, such as different headers or footers. You can modify the DITA OT files that install with oXygen, but we recommend you configure oXygen to use your own customized or specialized DITA OT.

To specify which conditional content to publish, you create a transformation scenario for the DITA map. In this scenario, you either select the conditions to publish, or you select a ditaval file you've already created.

As for supporting specific DITA components, oXygen handles DITA maps, conrefs, conditional processing, and specialization.

oXygen ships with Apache FOP, while XMetaL Author ships with both RenderX XEP and Apache FOP. XEP is a more robust XSL-FO engine than FOP.

DITA Studio

DITA Studio is an XML authoring tool specifically for authors comfortable with word processing programs (such as Microsoft Word). DITA documents are displayed as you see a document displayed in Word.

DITA Studio converts FO to a PDF file using FOP. You can configure the program to use other FO processors.

As with the other XML authoring tools, DITA Studio simplifies conditional processing. It creates the ditaval file for you after you select the filtering attributes. You don't have to remember the ditaval syntax.

You modify PDF file formatting in DITA Studio by customizing the XSL-FO provided with the DITA OT.

ePublisher 2009

ePublisher is a flexible document conversion tool integrated directly with the DITA OT. ePublisher is in a separate category from the other tools because you cannot author content in ePublisher; you can only import and convert it.

ePublisher transforms the XML into an FO file and then uses RenderX XEP to render the FO as a PDF file. The PDF file includes bookmarks and thumbnails.

As with the XML authoring tools, ePublisher comes with its own installation of the DITA OT. You have the option of using your own installation of the DITA OT by copying the OT into the ePublisher project directory. Then you can modify the XSL-FO to change the formatting in PDF files.

ePublisher automatically inserts the DITA map title in the PDF file header. To add other content to the headers or footers, you modify the XSL-FO to do so.

As with the XML and help authoring tools, ePublisher can be run from the command line to automate PDF output production.

As for supporting DITA components, ePublisher handles DITA maps, conrefs, conditions, and specializations. It's a likely contender for heavy-duty DITA users.

As with the previously reviewed editors, ePublisher is a powerful help authoring tool for DITA authors who are familiar with the DITA OT and XSL-FO, but modifying PDF file formatting will be challenging to the casual author who might not know XSL-FO. Many users will also find it frustrating that the formatting can't be modified in the ePublisher Style Designer. The program is flexible in terms of using customized or specialized DITA OT installations.

Help authoring tools with DITA import and PDF file conversion

This type of solution strikes a medium between the page layout solutions and the FO-oriented solutions. Help authoring tools typically produce pages with fairly generic styling, so don't expect magazine-quality layout. On the other hand, PDF file formatting is set up through CSS styles. This means that producing a pretty PDF file is much easier than in FO solutions and doesn't require hiring a programmer to develop the FO.

Tools in this category include Adobe RoboHelp and MadCap Flare.[2]

These solutions have the following characteristics:

- Page layout and formatting control is not an emphasis. These products are centered around web-based help, and the PDF file layouts reflect that quality of layout. On the other hand, general stylistic customizations are easier to make than with FO stylesheet changes or changes to page layout-oriented applications.
- The addition of custom features (such as inserting a title page based on metadata) are not generally practical with these products.
- Because these tools are designed for much more than just PDF file publishing, they also are about as expensive as page layout software—$900 to $1000.
- Both programs in this category can be run from the command line to automate PDF production.

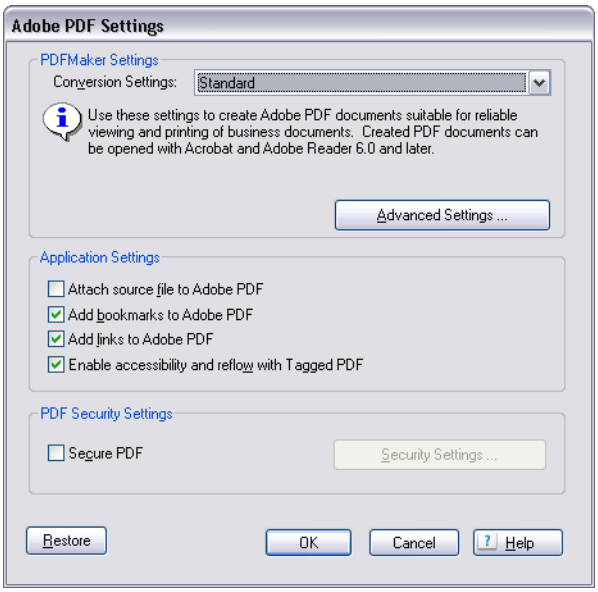
If you are looking for an out-of-box experience and you need cross-platform, web-based outputs in addition to PDF files, help authoring tools may be a more appropriate path than using page layout applications.

RoboHelp 8.0

RoboHelp is a feature-filled help authoring tool that has been updated to import DITA content, which is then converted to an intermediary format—XHTML. The imported content is converted to a PDF file via Adobe PDF. Because Adobe develops both Adobe PDF and RoboHelp, the PDF file options are tightly integrated with RoboHelp. For example, you can embed bookmarks in the PDF file and enable accessibility (Figure 3 on page 10).

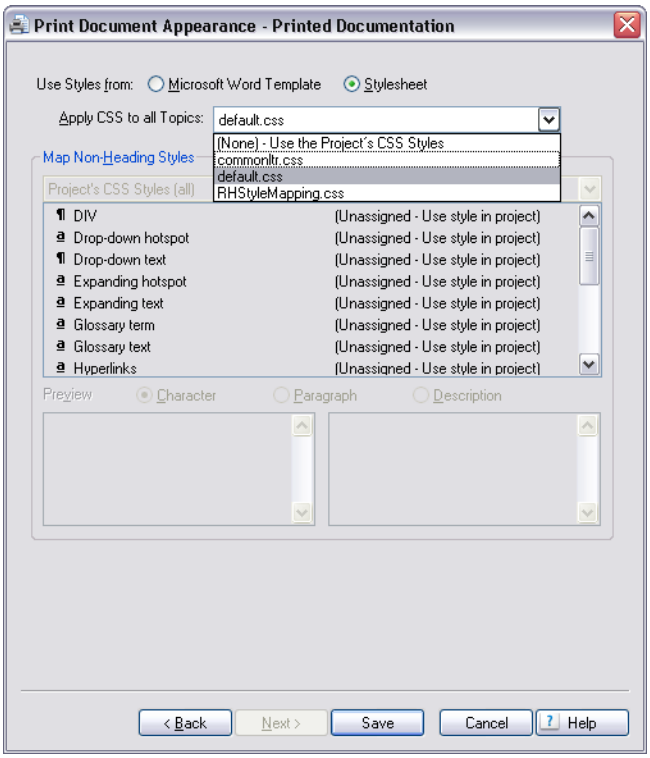
2. Author-it supports DITA authoring and PDF file conversion; however, you cannot import DITA files solely for conversion to PDF files. For this reason, the program is not covered in this paper.

Figure 3: PDF file options in RoboHelp



As in Flare (which you'll read about later), you can create and edit Cascading Stylesheet (CSS) files in the program or assign a stylesheet created in another program. You then assign a specific stylesheet to the PDF output. This lets you format a PDF file in a serif font, for example, and HTML in a sans serif font, if you'd like. Figure 4 shows the selection of a CSS for the PDF document.

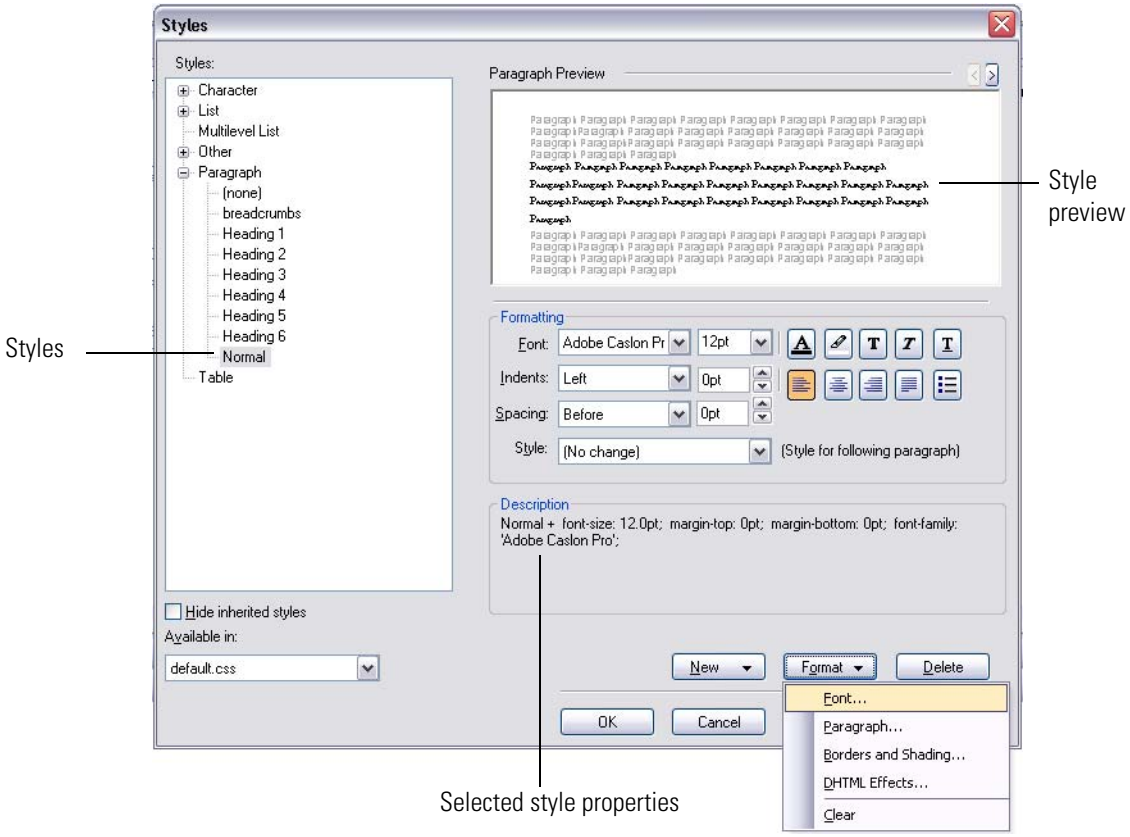
Figure 4: Stylesheet options in RoboHelp



The RoboHelp CSS formatting isn't as refined as in Flare. In both programs, you can modify and create new CSS styles for most types of text in the CSS editor. But in RoboHelp, the Normal CSS style is applied to headers and footers. To format the headers and footers differently from body text, you apply font properties directly to the text on the master page; you have no access to a CSS style applied specifically to the headers and footers.

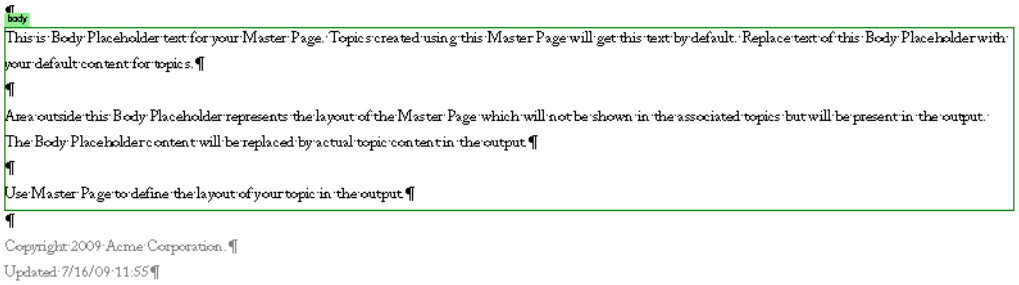
The CSS editor in RoboHelp offers fewer options than in Flare. As shown in Figure 5, the editor is similar to the style editor in Word, with font, paragraph, border and shading properties.

Figure 5: CSS Editor in RoboHelp



You can create a master page with breadcrumbs, a mini-TOC, and other components and apply them to the title page. Figure 6 shows a placeholder for the body page content on the master page. The copyright statement and modification date are inserted in the footer. You apply the master pages to individual topics or to all topics during conversion to PDF.

Figure 6: Master page view in RoboHelp



In the PDF file headers, the book title (on even pages, left-aligned) and the chapter name (on odd pages, right-aligned) are displayed. You can also add your own content to headers and footers (for example, a copyright statement or modification date).

As for supporting specific DITA components, RoboHelp recognizes DITA maps, conditions (including selecting your own ditaval file), specializations, and conrefs.

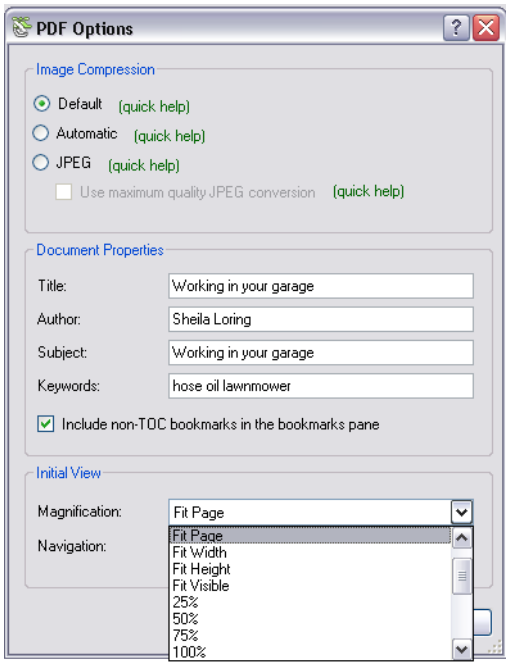
RoboHelp is a good alternative for users who want to take full advantage of Adobe's PDF integration. With the advanced PDF file options, you can create press- or online-ready PDF files. The master page feature makes it fairly easy to customize the page layout in PDF files. We would like to see more control over CSS formatting. Flare offers more flexibility in this regard. If you want to edit the XML, look for an XML authoring tool instead.

Flare 5.0

Flare is a powerful help authoring tool that supports a number of output formats. MadCap introduced DITA support in version 5. As in RoboHelp, you import DITA files into the program, and the files are converted to an intermediary format—XHTML. Flare also recognizes DITA maps, conrefs, and conditional processing. It does not, however, support specializations.

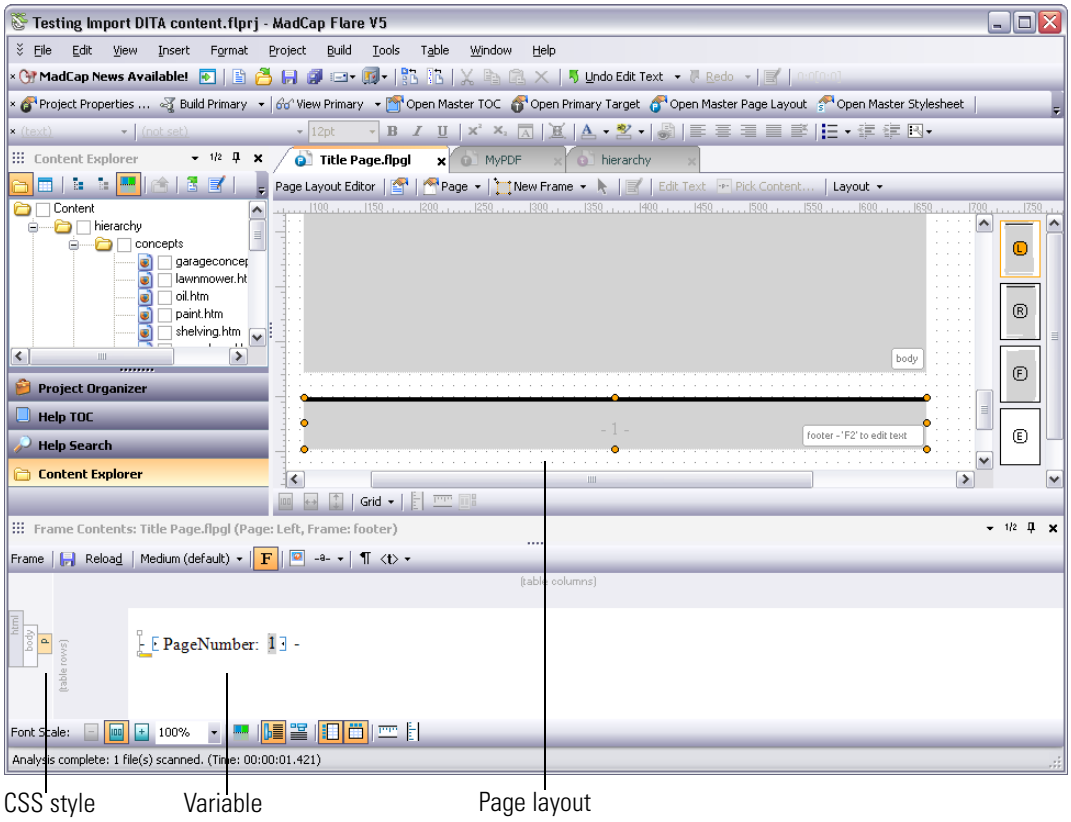
Flare uses an XSL-FO process to convert topics to an FO file and then a PDF file. You can set the image compression, document properties (such as author name and keywords), and initial view through the PDF Options dialog box (Figure 7). These PDF file settings are a bit more limited than in RoboHelp, which is fully integrated with Adobe PDF.

Figure 7: PDF file options in Flare



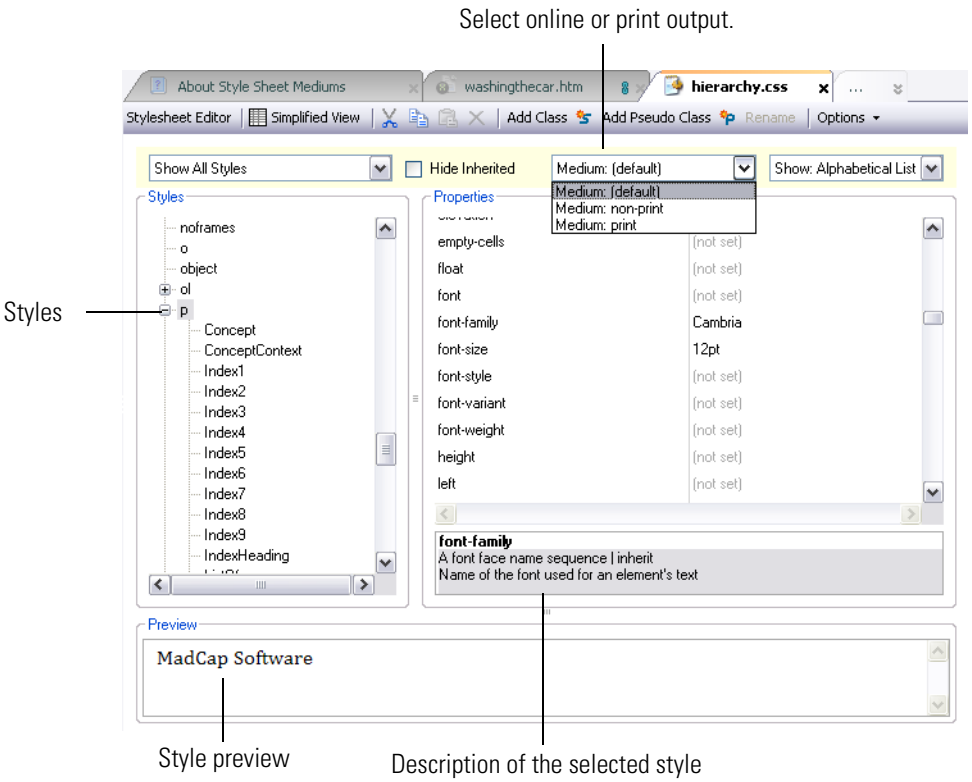
In Flare, you can set up a specific page layout for PDF output (Figure 8). For example, the PDF file may have a title page and different headers and footers from the HTML. In the page layout editor, you can insert variables for the book title, heading, page number, assign a CSS class, and more. You assign page layouts to specific topics, just like in RoboHelp.

Figure 8: Flare page layout editor



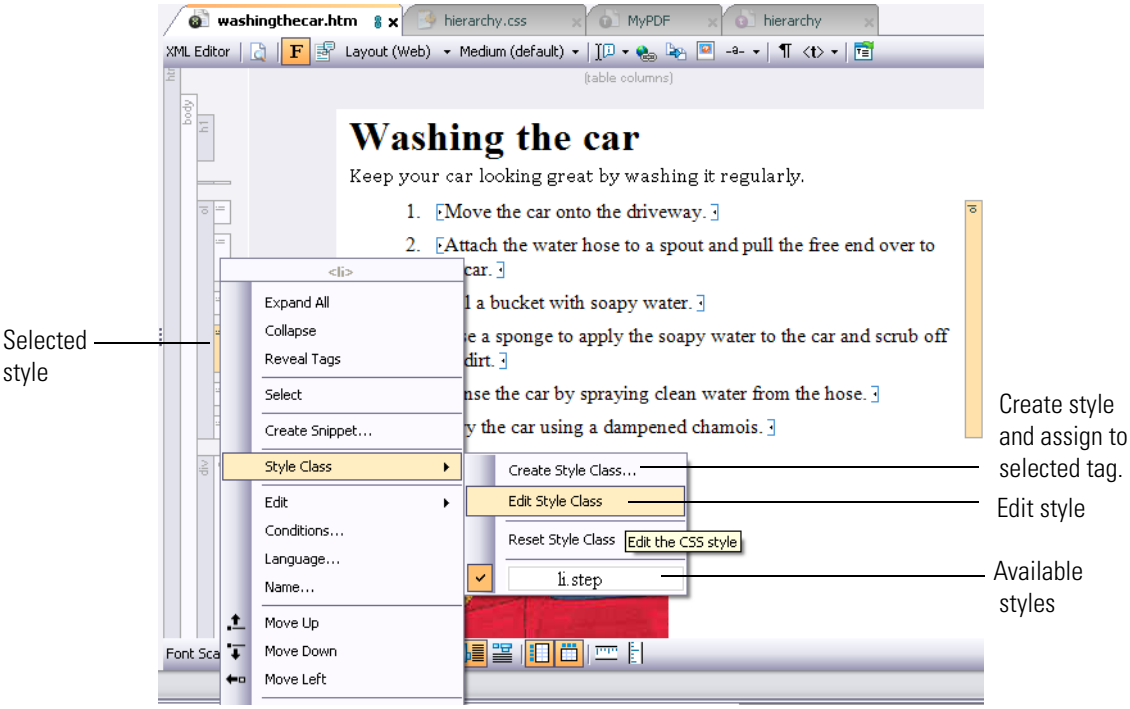
Flare also lets you create and edit CSS styles (Figure 9 on page 14). You have control over everything from azimuth (the horizontal angle of the text) to the z-index (the stacked order of a positioned element). The CSS can be customized for a single project or for an output type. So if you want all PDF file output to be formatted the same, you assign a CSS to the entire output type. (The CSS file can also be created in another tool and imported into the project.)

Figure 9: Flare CSS editor



You assign or select a style by right-clicking on the HTML tag inserted in the document (Figure 10). All valid CSS styles for the selected tag are displayed. For example, in Figure 10, the "li" tag was selected, and only the "li step" class can be assigned to the tag.

Figure 10: Assigning a class in Flare



Flare makes it easy to format PDF file output with CSS. It's a good choice for users who don't want to rely on a programmer. Flare won't let you author DITA content, but you can edit the intermediary XHTML. Unlike RoboHelp, Flare doesn't support specializations of the DITA OT.

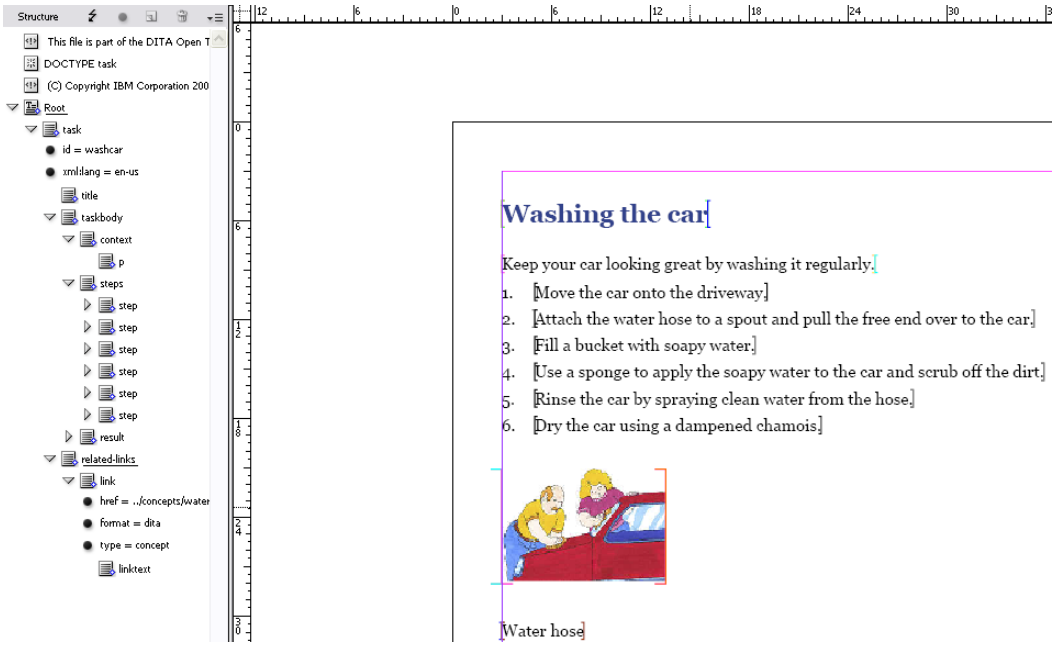
- Flatten the structure (including expanding topicrefs and conrefs).
- Modify image and table structures.
- Provide formatting information through the aid namespace (aid:pstyle=”Body”).
- Pre-process conditional elements.

These transforms are available for free on the Scriptorium wiki at http://wiki.scriptorium.com/tiki-index.php?page=DITA+to+InDesign. These scripts are proof of concept and not for production.

2. Create a new InDesign document and make sure the file has the paragraph styles you specified in the namespaces. In the previous example, the paragraph style is “Body.”
3. Import the XML.
4. Process the cross-references.
5. Save the file as a PDF document.

Figure 11 shows a DITA document imported into InDesign. The Structure palette shows the elements in the document.

Figure 11: A DITA document imported into InDesign



Once you go through these steps, you have the benefits of a high-end page layout tool that will give you many more layout options than XSL-FO can provide.

In addition to not supporting DITA maps and conditional processing, InDesign doesn’t support specializations or conrefs. InDesign is clearly not optimized for DITA. If precisely formatted PDF files are most important to you, and you’re willing to write scripts (or adapt Scriptorium’s scripts), InDesign might be the tool for you. FrameMaker, which you’ll read about next, provides a more seamless way to produce PDF files from DITA.

NOTE: Quark XPress also can convert DITA to a PDF file, but the process must be scripted as in InDesign.

FrameMaker 9

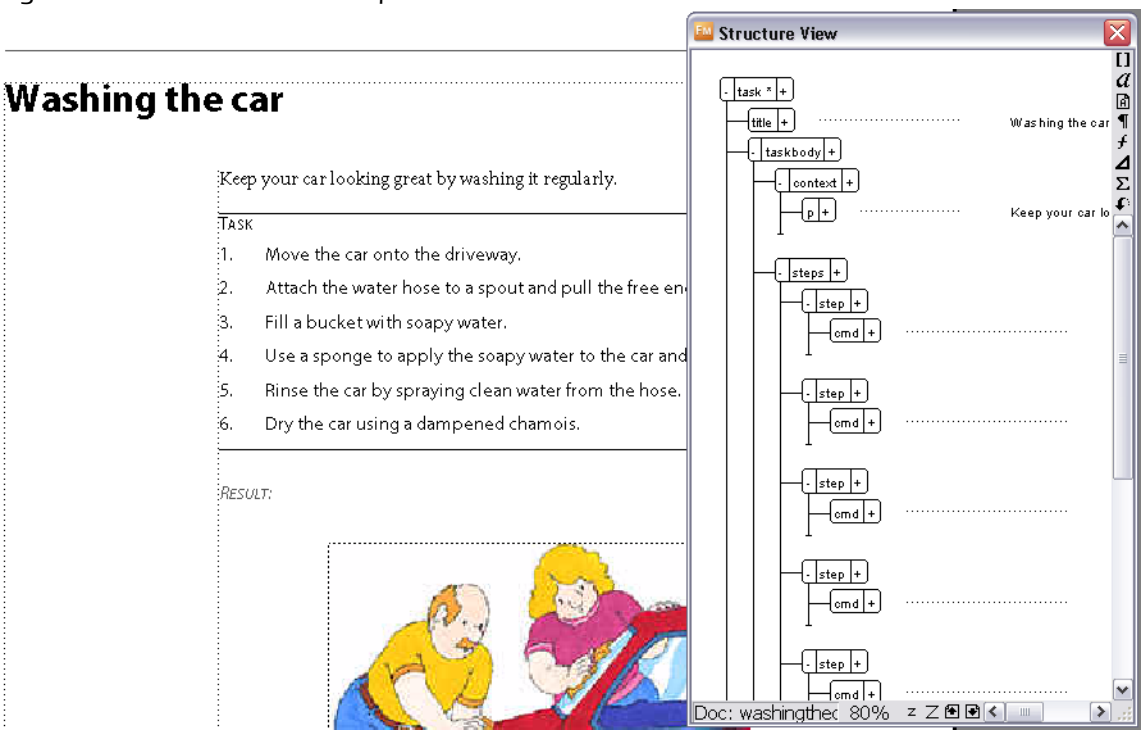
FrameMaker is in the space between the high-end layout solutions and the XSL-FO based solutions. While FrameMaker does have excellent layout capabilities, it is not as flexible when handling complex layout with multiple text streams as InDesign. However, its handling of XML, and DITA in particular, is better than InDesign because scripting is not required for items such as cross-references, images, and tables.

The basic conversion steps are as follows:

1. Open the DITA content into FrameMaker. FrameMaker provides a structure application, which defines rules for importing, exporting, and formatting content. The structure application is run, and the content is converted to a FrameMaker document.
2. Save the file as a PDF file.
3. If necessary, edit the structure application to customize formatting or import and export rules.

Figure 12 shows a DITA document imported into FrameMaker. The Structure View window indicates the hierarchy of elements inserted in the document.

Figure 12: *A DITA document imported into FrameMaker*



The advantages to using FrameMaker for rendering PDF files from DITA are significant:

- FrameMaker formatting is reasonably easy to modify.
- FrameMaker offers broad support for Acrobat PDF file conversion.

- Unlike InDesign, FrameMaker handles DITA maps.
- The process of importing DITA to FrameMaker works almost seamlessly.

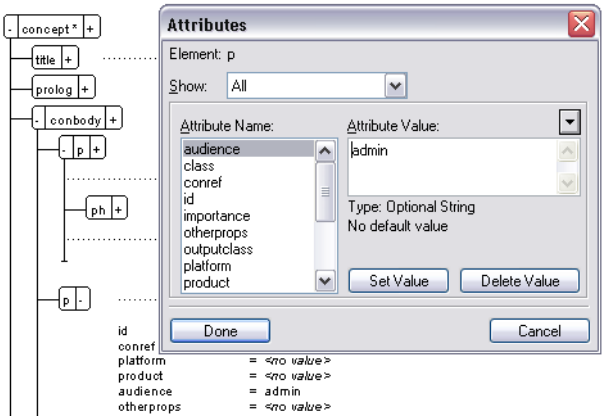
Most XML element relationships can be preserved inside the structured FrameMaker environment. What you edit in structured FrameMaker is not XML, but a structured equivalent. Thus, there is at least the potential for being able to round-trip content from XML to FrameMaker and back to XML in the original structure.

For DITA, round-tripping is problematic in FrameMaker. For example, column widths in some tables are not round-tripped. A more pragmatic solution might be to perform editing of DITA in an XML editor and then use structured FrameMaker to import the DITA content and create the PDF file. Still, you'll need Leximation's third-party DITA-FMx solution to optimize the files (for example, to strip XML padding from pretty-printed files, round-trip table column widths, support index terms as markers and elements, and support "pgwide" tables). DITA-FMx completely replaces FrameMaker's DITA options; you cannot have both FrameMaker's DITA OT options and DITA-FMx installed together. DITA-FMx also comes with its own structure application, which you can also customize if necessary.

Customizing the layout for imported DITA content involves modification of FrameMaker's (EDD) file, FrameMaker styles, and other parts of FrameMaker's structured application for DITA. Some expertise in FrameMaker is needed for doing this.

FrameMaker supports the processing of conditional content. In FrameMaker 9, you can specify an existing ditaval file for PDF file or print output. To select the ditaval for other types of output, you'll need to install DITA-FMx.

Figure 13: *Assigning a condition in FrameMaker*



Note that you must download both the DITA OT and the Adobe DITA OT plug-in to get DITA support in FrameMaker. During the DITA OT set up, you set many environment variables that are automatically set when you run the DITA OT alone. These environment variables might clash if you're using the DITA OT alone for any other project on the same computer.

DITA-FMx is also a separate installation and completely replaces the FrameMaker DITA options listed in the drop-down menus.

Spending time to learn how to modify the DITA OT or hiring someone to do so would be more expensive than using FrameMaker. More advanced options for components such as indexing and ditaval filtering require the purchase of DITA-FMx, which is priced at $185 per license, with volume discounts available.

XML publishing tools

Some programs are integrated with the DITA OT but use their own internal conversion process to convert DITA to a PDF file rather than going through an XSL-FO processor. XML Professional Publisher and Arbortext Publishing Engine fall in this category.

These tools are known to be more expensive than any other tool mentioned in this paper; therefore, they're typically only practical for enterprise-level publishing.

XML Professional Publisher

XML Professional Publisher (XPP) is the SDL XySoft dynamic publishing engine for automated print and PDF file publishing. XPP produces high-volume, intricately formatted publications. This publishing engine includes specialized features to accommodate commercial publishers, financial printers, journal producers, and in-house technical publishing departments that produce newsletters, books, magazines, loose-leaf manuals, or catalogs.

XPP has an interface called XyView that lets the user edit documents and inspect or override formatting decisions. The styles in a document can be updated so that a particular block is processed in a specific way and the changes don't apply to the entire document. More complex formatting can be accomplished with Perl scripts.

XPP is integrated with the DITA OT through a plug-in SDL XySoft provides. The plug-in applies ditaval filtering, creates an XML file, and then applies styles to produce the PDF file. XPP plays the role of an FO processor but uses proprietary FO language instead of FO.

Publications can be produced in batch mode, through an interface, or through the Web Services API. XPP runs on Windows 2003 or 2008 server, Windows XP, Windows Vista, Red Hat Linux, Solaris, and AIX.

SDL XySoft wouldn't provide pricing information. Visit SDL XySoft's website for more information: http://www.xyenterprise.com/products/xpp.html.

Arbortext Publishing Engine

Arbortext Publishing Engine is a high end web-based publishing server from Parametric Technology Corporation (PTC). The Publishing Engine pulls content from various content management systems or file systems and a host of file formats (including Microsoft Word, Adobe FrameMaker, HTML, text, and PDF files). PTC offers two versions—a server that only generates PDF files, and a full-featured version, which converts content to HTML, HTML help, wireless, and more.

The tool uses its own publishing engine to create PDF files. Extensions are supported for index generation, change bars, column-wide footnotes, and more. The publishing engine can be called from other applications using Perl, C, Python, and ANSI C. It runs on Windows, Solaris, and UNIX.

PTC wouldn't provide pricing information. Visit PTC's website for more information: http://www.ptc.com.

Conclusion

In summary, we group tools according to solution type and rank them on a scale of 1 to 6, where 1 is the most favorable score.

Tool	Page layout sophistication	Automation*	Cost**
The DITA OT with the open-source XSL-FO processor FOP	4	1	1
The DITA OT with a commercial XSL-FO processor (XSL Formatter and XEP)	3	1	5
XML authoring tools (XMetaL Author, oXygen, and DITA Studio)	5	1	3
Conversion tools (ePublisher)	5	1	4
Help authoring tools (Flare and RoboHelp)	6	1	3
Page layout software (InDesign and FrameMaker)	1	2	2
XML publishing tools (XML Professional Publisher and Arbortext Publishing Engine)	2	1	6

* Automation is based on command-line operations. Transformations can be run from the command line in all programs but InDesign and FrameMaker, the latter of which requires a plug-in such as FrameScript to provide this functionality.

** Actual cost is based on the time devoted to customizing the FO. This ranking takes into consideration minimal customizations.

Based on the information summarized in the table, evaluate the following factors before selecting your DITA-to-PDF file process:

- If automated production is most important, consider the DITA OT and an XSL-FO processor. You can use a free processor, FOP, but FOP isn't as robust as other processors.
- If you want to author DITA and produce PDF files through an interface, consider an XML authoring program, such as XMetaL, oXygen, or DITA Studio; however, these programs don't allow for the sophisticated typography and page layout made possible in page layout programs such as FrameMaker and InDesign.
- To simplify the formatting process, consider a help authoring tool (such as Flare or RoboHelp) that lets you format PDF file output with CSS styles. Realize that these tools don't provide as much fine control over page layout as FrameMaker, InDesign, XPP, and Arbortext Publishing Engine.

- If hand-tweaking the formatting is top priority, consider a page layout program such as InDesign or FrameMaker; however, you'll need to run import scripts (in InDesign) or purchase the add-on DITA-FMx for FrameMaker for increased DITA functionality.

In the end, you might choose a combination of methods. For example, you could combine an XML authoring tool (for writing) and XPP or Arbortext Publishing Engine (for complex formatting options and conversion through an interface). You should carefully weigh the alternatives and trade-offs.

About the authors

Sheila Loring creates systems that produce and repurpose content. These days, most of her work is XML based, and many systems are based on the Darwin Information Typing Architecture (DITA) and the DITA Open Toolkit. Sheila modifies and extends XSLT, CSS, and Ant files in the DITA OT to produce deliverables that exactly match customer requirements.

Sheila coauthored *Publishing Fundamentals: FrameMaker 7* (originally published as *FrameMaker 7: The Complete Reference)* and *The WebWorks Publisher Cookbook*. She also contributes to the Scriptorium blog, Palimpsest.

David Kelly has worked since 1978 as a technical writer, publications coordinator, documentation manager, software project manager, program manager, and technical consultant. Currently, he manages various Scriptorium projects and spends time working on new documentation automation solutions.

Always interested in the use of new tools to improve documentation processes and quality, David has devised several enhancements to the DITA Open Toolkit for Scriptorium's customers, from processing trademarked terms to translating Microsoft Word documents to DITA output. He also contributes to the Scriptorium blog, Palimpsest.

About Scriptorium

Scriptorium Publishing provides expert advice on how to develop, deploy, and manage content. Our typical customer has thousands of pages of information, which needs to be delivered in print, PDF files, HTML, and other media, often in dozens of languages. Our mission is to automate formatting and production tasks, usually through XML technologies, so that authors can write more efficiently.

Our consultants have experience in traditional publishing workflows, including typesetting, book design, copyfitting, and production editing. This understanding influences our approach to creating state-of-the-art publishing systems with modern tools and technologies, such as XML, HTML, DITA, the DITA Open Toolkit, XSLT, XSL-FO, FrameMaker, Ant, Perl, FrameScript, Flash, InDesign, XMetaL, oXygen, and many more.

Our customers include federal and state government as well as companies in defense, consumer electronics, telecommunications, health care, pharmaceutical, and other industries.

If you are facing a difficult publishing challenge, we want to hear from you. Contact us at info@scriptorium.com or 919-481-2701 x105.

Scriptorium Publishing is based in the Research Triangle area of North Carolina and has been in business since 1997.