



The What, Why, How, and Where of DITA

BY ROB STEINER, *Senior Member*

If you're like most technical communicators, you've seen the acronym DITA (Darwin Information Typing Architecture) discussed in forums like TECHWR-L, noticed STC teleconferences on the subject, and caught a passing reference in *Intercom* articles.

And if you're like most technical communicators, you're still pretty fuzzy on what DITA actually is.

This article will provide a starting point from which you can begin your own research on DITA. It will cover *what* DITA is, *why* technical communicators should care, *how* to implement DITA, and *where* to find free resources on the Web.

What Is DITA?

DITA is an XML-based information architecture designed specifically to organize and present technical content. It contains a set of document type definitions (DTDs) and schemas—in other words, XML “templates”—that support topic-based content development. DITA enables you to structure your content so that the same topics can be reused in several different formats, like online help or minimalist user guides.

Brief History

In 1999, IBM decided to organize and manage its millions of pages of technical documentation into a single-source information architecture that could

be shared with, and recognized by, other data-owning communities (e.g., partner companies and intra-company organizations).

The company came up with DITA in 2000. It was an XML-based documentation structure designed to maximize information interchange between multiple organizations, be tools-agnostic, and be extensible so that it could be adapted to any organization's specific needs.

In March 2004, IBM donated DITA to the Organization for the Advancement of Structured Information Systems (OASIS). OASIS now manages and updates the DITA specifications. The most recent update, DITA 1.1, was released for public review on March 6, 2007.

DITA Concepts

Several basic concepts lie at the heart of DITA.

It's XML-based: XML enables you to semantically describe the information between XML tags. For example, while HTML uses tags like <h1>, <h2>, and <p> to describe the structure of a document (Heading 1, Heading 2, and Paragraph, respectively), the tags say nothing about the content. XML, however, can have tags like <title>, <obj>, and <step> to describe a process's *title*, *objective*, and *steps*. DITA takes the best of XML and puts it into a set of ready-to-use DTDs and stylesheets.

Organizes content by topics: Topics are at the core of DITA content. The most basic DITA topic types are *tasks*, *concepts*, and *reference* items. Topics are short enough to describe a specific task or subject, yet long enough to make sense on their own. DITA topics can be gathered and arranged into specific deliverables, such as online help systems or user manuals.

Uses topic maps to define which topics are in a deliverable: Topic maps are similar to a table of contents. Without a topic map, all you have are a bunch of chunked topics without any logical organization. With DITA, you can create topic maps specific to online help systems, user guides, and training materials, all using the same DITA topics.

Makes content reusable: DITA content enables faster development of technical documentation by making each topic reusable, reducing the need to copy information into multiple deliverables. In other words, you write a topic once and then use topic maps to add it to your online help, knowledge base, or user guides.

Allows for specialization: Specialization is the process of customizing DITA to fit your organization's particular needs. So rather than using DITA's generic *task*, *concept*, and *reference* topic types, you can create your own topic types and the elements associated with them.

Offers conditional text tags for each deliverable: Say you want to generate a one-page job aid with only the instructions for a certain software

task, but don't want to include all the screen shots. You can embed those screen shots within DITA tags that allow the screen shots to display only when you are using a topic map for "software manual," not a topic map for "job aids."

DITA Structure

As previously stated, the basic structure of DITA content is the *topic*, and a *topic map* organizes topics into separate deliverables.

For example, the chart in Figure 1 shows the relationship between topic maps and topics. Topic maps contain many topics, while each topic contains your XML-tagged content. Each topic contains tags specific to its topic type (*task*, *reference*, or *concept*).

DITA Topics and Markup Examples

If you know the basic structure of HTML and XML, you already know how DITA-tagged content will look.

Generic topics: The generic topic structure is used for topics that do not have a specific information type.

```
<topic id="topic">
<title>Chess</title>
<body>
<p>Chess is a strategic, two-player
game played on a 64-square check-
ered board.</p>
</body>
</topic>
```

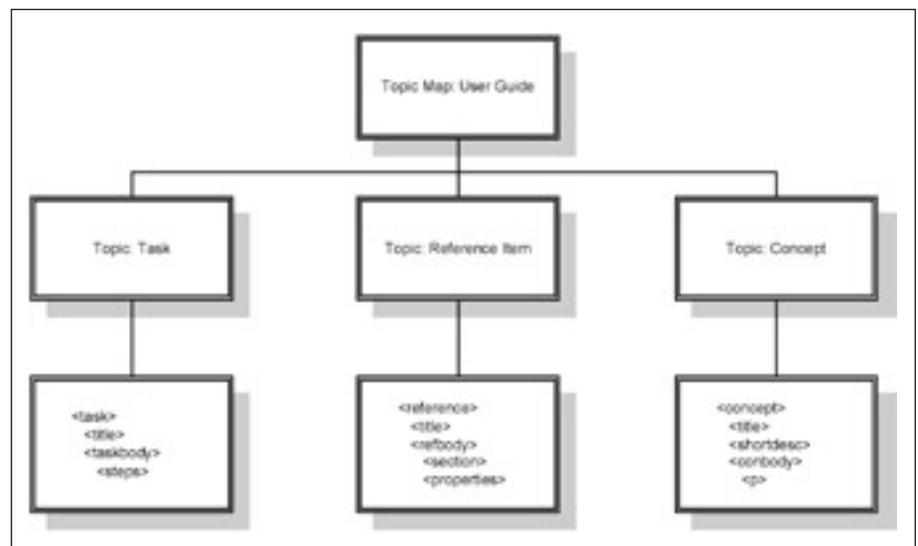
Task topics: Task topics describe the steps of a specific task. These topics include tags that identify the task's context, prerequisites, steps, expected results, and next steps.

```
<task id="movingpieces">
<title>Moving a chess piece</title>
<taskbody>
<context>The only way to win is if you
move your pieces.</context>
<steps>
<step><cmd>Grab the piece you want
to move.</cmd></step>
<step><cmd>Move the piece to the
desired location.</cmd></step>
</steps>
</taskbody>
</task>
```

Concept topics: Concept topics answer the question, "What is?" They provide background or overview information for tasks or reference topics. Concept topics are often used in glossaries and in examples, or they may even contain graphics.

```
<concept id="chessintro">
<title>History of chess</title>
<shortdesc>Chess is a strategic board
game invented in India sometime dur-
ing the sixth century.</shortdesc>
<conbody>
<p>Each side has sixteen pieces:</p>
<ul>
<li>King</li>
<li>Queen</li>
```

Figure 1. The basic structural element of DITA—the topic—is organized by topic maps, with each topic containing specific, stand-alone content.



```
...
</conbody>
</concept>
```

Reference topics: Reference topics describe the regular features of a set of things, such as programming languages, facts about a product, catalogs, or even recipes.

```
<reference id="refsample">
<title>The pawn</title>
<refbody>
<properties>
<property>
<proptype>Pawn</proptype>
<propvalue>1 point</propvalue>
<propdesc>Each side has eight pawns,
which start out arranged on the sec-
ond rank on opposite sides of the
chess board.</propdesc>
</property>
</properties>
</refbody>
</reference>
```

Topic map: A topic map includes all topics that you want to include in a particular deliverable and in what order they should be displayed. Topic maps are what keep DITA topics context-free, enabling their reuse across multiple deliverables. A topic map might look something like this:

```
<map title="Getting Started With
Chess">
<topicref href="history.xml">
<topicref href="invention.xml"/>
<topicref href="moderngame.xml"/>
</topicref>
<topicref href="pieces.xml">
<topicref href="king.xml"/>
<topicref href="queen.xml"/>
<topicref href="bishop.xml"/>
<topicref href="knight.xml"/>
<topicref href="rook.xml"/>
<topicref href="pawn.xml"/>
</topicref>
</map>
```

Why Should Technical Communicators Care about DITA?

With all the buzz and hype surrounding DITA, how can you know whether DITA is right for your organization?

You Might Consider DITA If...

Your content is online. If so, DITA might be a good fit for you, more so if your content is already in structured XML. The basic DITA topics of *concept*, *task*, and *reference* are similar in structure to today's standard help systems, creating a potentially smooth conversion.

You exchange technical documentation with other groups that use DITA. If so, DITA is certainly for you. Even if you do not exchange such information now, seriously consider whether you might do so in the future. If so, DITA will likely be the default architecture for such exchanges.

You will have a DITA "enforcer." Converting content to DITA is a waste of time if your writers will not follow DITA writing styles when developing new content. To enforce compliance, it's helpful to have someone review the output of your writers to ensure they are following the standards. Whether this person is a manager or editor, he or she must have the support of upper management in case any disputes arise with your writers.

You have a content management system (CMS) or plan to purchase one. Moving your content to DITA without a CMS is certainly doable, but quite dangerous. For one thing, you would not have the change controls that a CMS can provide. In other words, anyone can go in and change your content without your knowing it. And what happens when two people unknowingly work on the same topic, one copying over the other's changes? A CMS can be designed so that it forces your writers to conform to DITA writing styles, all while tracking any changes to topics and even generating the deliverables themselves.

DITA Might Not Be for You If...

Your content is not topic-based. DITA is designed for structuring content that is modular, not sequential. For example, you would not want to convert a novel to DITA since you can't understand Chapter 10 without first reading the previous nine chapters.

Your content is print-based. While it is within DITA's capabilities to structure print-based books, marketing materi-

als, product sheets, etc., all such output would be quite minimalist without some degree of DITA specialization. You will have to evaluate whether the benefits of creating new topic types and elements would be worth the time and effort.

You do not have the resources to convert your existing content to DITA. Converting existing content to DITA is not for the faint of heart. It requires executive commitment, time, and a critical examination of what your content is, how it's delivered, who uses it, and how it gets updated. You might have to rewrite and restructure your content so that it conforms to DITA. If you do not have the resources for this task, you should either wait for the development of mature software tools that can ease the process, or stick with the content management solution that is working for you now.

How Do You Implement DITA?

Say you've decided to take the plunge and convert your existing content to DITA. Where do you start? The answer to that question is different for every organization, but here are some basic steps that should help get you started with your own conversion process.

Step One: Learn about DITA

You've already taken this step by reading this article. Later on I'll provide a list of Web sites that will give you even more detailed information.

Step Two: Evaluate your existing content

Look over your existing content and determine whether it would lend itself to DITA conversion. Online content is ideally suited for the basic DITA topic types (*task*, *concept*, and *reference*), but print-based materials can also be converted to the basic DITA topic types if they are minimalist and do not require extensive specialization.

Step Three: Do you need to specialize?

Using the basic, built-in DITA topic types is the easiest and most cost-effective way to implement DITA. But what if your content does not fit into those topic types? Existing DITA DTDs and stylesheets can be customized to

fit your organization's content needs. Though this requires a lot of planning and is initially labor-intensive, the end result will be a highly optimized system that satisfies your organization's specific requirements.

Step Four: Select a tool

Technically speaking, you can use a simple text editor like *Notepad* to create DITA topics...but then you'd likely have a writer revolt on your hands. For DITA to be enthusiastically embraced by your writers, you need a software tool that enables them to focus on the writing, while leaving the XML tagging to the software.

Any XML-editing software can support DITA, since DITA is a set of XML tags, stylesheets, and DTDs. What you need to determine is whether a tool adds any useful functionality specific to DITA development. Look for tools that offer free trial versions, download the tools, and put them through rigorous testing. Make sure you and your authors are comfortable with the tool before you decide to plunk down a lot of money for it.

In addition to an XML authoring tool, you may want to consider a CMS, if you do not already have one. Best practices indicate that DITA implementation without a CMS in which to store topics puts change control at risk and hampers reusability, which is one of the major features of DITA.

For a list of the most popular XML authoring tools and CMSs, see dita.xml.org/products-services.

Step Five: Create a conversion plan

Start off with the "easy" content that is already topic-based, such as online help. Once you've converted your online content and tested its output, you can start on any print or specialized online materials you want to convert.

Step Six: Convert content in waves

This is where the time-consuming tasks of rewriting and restructuring your content begin, since much of your content will be in multiple formats (*Word*, *FrameMaker*, etc.) that probably have inconsistent styles and structure. It's best

to roll out converted content in waves to minimize the impact on existing projects (since project work never stops!) and to work out the kinks that will inevitably pop up with each wave so that you can use those solutions for future waves.

Step Seven: Create new content

This step can happen simultaneously with Step 6, since you'll likely continue to receive writing projects even as your conversion process progresses. This is where the writers on your team must focus on using the new styles—topic-oriented, modular—and entering their output into the CMS, if you have one. This is also where the DITA "enforcer" becomes one of the most important members of the team, especially in the beginning when your writers are learning the new system.

Where Can I Find More Information on DITA?

As you might expect, the Web contains vast amounts of information on DITA. A simple keyword search in any search engine demonstrates this. The trick is in knowing which sources are valuable. I've provided a few that should get you started.

DITA.XML.org: This site contains all the core DITA documentation you need to get started. You can find case studies, DITA products and services, and user forums to help with your conversion. dita.xml.org

DITA Open Toolkit: The DITA Toolkit is the complete, open-source set of DITA DTDs and schemas. Any software tool that claims to support DITA should have the Toolkit built into it. If, however, you want to build it into your existing CMS, the Toolkit is free and can be downloaded from sourceforge.net/projects/dita-ot

OASIS: OASIS is the nonprofit body that defines, maintains, and promotes the DITA standard. The current standard is DITA 1.0, but OASIS released DITA 1.1 for public review in March 2007. www.oasis-open.org/committees/dita

IBM: IBM initially developed the DITA standard in 2000, and then donated it to OASIS in 2004. The IBM site still contains several technical papers and FAQs

on DITA. www-128.ibm.com/developerworks/xml/library/x-dita3/index.html

ContentWrangler.com: Scott Abel's blog is an excellent source of information on DITA, content management systems, and all things technical writing. www.thecontentwrangler.com

Successful Implementation Requires Effort

The market is still trying to figure out what DITA means to technical communication and how it can be used cost-effectively. As many experienced users have noted in the DITA forums, DITA is not a magic wand that will make all your documentation headaches go away. A successful implementation requires commitment, time, and complete buy-in from your writers and executive management. Do your research, test the tools, and make the decision that's best for your company. 📌

SUGGESTED READINGS

ContentWrangler.com.
www.thecontentwrangler.com

Day, Don, Michael Priestley, and David Schell. *Introduction to the Darwin Information Typing Architecture*. www-128.ibm.com/developerworks/xml/library/x-dita1

_____. *Frequently Asked Questions about the Darwin Information Typing Architecture*. www-128.ibm.com/developerworks/xml/library/x-dita3/index.html

DITA World. www.ditaworld.com

DITA-XML.org. dita.xml.org

Rob Steiner (robsteiner.com) is a Web communications specialist with Sears Holdings Corporation in Troy, Michigan. He specializes in writing Web content and developing user interfaces. He is a senior member of the Southeastern Michigan Chapter STC, and can be reached at robsteiner01@gmail.com.

Discuss this article
online at [stcforum.org/
viewtopic.php?id=879](http://stcforum.org/viewtopic.php?id=879)

