

What is DITA and Why Should You Care? (Sep 10)

By Chris Benz

September 27, 2010

“The key to understanding how DITA works is to understand how DITA uses topics, maps, and output formats. I will describe each of these in detail, but here's the big picture: You develop your content in DITA topics, use DITA maps to specify which topics go into which deliverables, then process those maps to DITA output formats to generate your final deliverables.”

Many of today's instructional developers face a significant dilemma.

Learners have minimal time to comprehend and effectively use complex products and systems. To drive time-efficient learning experiences, developers must provide high-quality training content, customized to specific learner roles and delivered in a timely manner. At the same time, many instructional development budgets are shrinking. In short, learners have less time and money to learn what they need to know, and developers have less time and money to deliver what those learners need.

One way developers can address this dilemma is to become more efficient at reusing content. For many developers, the best way to achieve that efficiency will be the Learning and Training Content (L&TC) Specialization, soon to be released in version 1.2 of the Darwin Information Typing Architecture (DITA) standard.

“The *what?*” some readers are surely asking. Before I answer that, let me give you a taste of the “why.” In a test project for the DITA L&TC Specialization, a team at IBM studied content reuse in an existing training course. They discovered that 50% of the course content had been copied from the product documentation. Using the Specialization, they were able to automate much of that reuse, not only avoiding the cost and potential errors of manual copying and pasting, but also providing an efficient way to synchronize content updates between product documentation and training materials, and saving on the cost of translating essentially the same content twice. Does this sound like something worth learning about?

In this article, I will provide a description and brief history of DITA, explain how DITA facilitates efficient and flexible reuse across various types of content deliverables, and discuss how the L&TC Specialization focuses the benefits of DITA on training deliverables.

What is DITA?

DITA is an XML-based open standard for structuring, developing, managing, and publishing content. IBM originally developed DITA to more efficiently reuse content in product documentation. In 2004, IBM donated their DITA work to the Organization for the Advancement of Structured Information Standards (OASIS) for further development and release to the public. OASIS formally approved the DITA 1.0 specification in 2005, and the DITA 1.1 specification in 2007. OASIS expects to formally approve DITA 1.2, including the L&TC Specialization, by the end of 2010.

Sidebar: Why Darwin?

Many people new to DITA ask, “Why the ‘Darwin’ in ‘Darwin Information Typing Architecture?’” As part of his theory of natural selection, the naturalist Charles Darwin noted that plants and animals inherit traits from their parents. Likewise, many elements in DITA inherit *attributes* from parent elements. Art reflects nature.

DITA has gained widespread adoption in the technical documentation world, in companies such as Cisco, IBM (of course), Nokia, and Oracle. But DITA adoption isn't exclusive to high tech. Boeing and the U.S.

Veterans Health Administration both participate on DITA committees, and I personally helped migrate the product documentation for ITT Fluid Technologies (manufacturer of pumps and valves) into DITA. DITA adoption has lagged in the training development world, however, which is not surprising given DITA's initial focus on technical publications and not instructional content.

DITA itself is not a tool, but many tools that support DITA exist. The DITA Open Toolkit (DITA-OT) is open source, and provides a solid starter set for working with DITA. Many DITA developers, however, find it easier and more productive to use commercial DITA tools. You can learn more about the DITA-OT and other DITA tools at <http://www.ditaworld.com/#tool>.

How does DITA work?

The key to understanding how DITA works is to understand how DITA uses *topics*, *maps*, and *output formats*. I will describe each of these in detail, but here's the big picture: You develop your content in DITA topics, use DITA maps to specify which topics go into which deliverables, then process those maps to DITA output formats to generate your final deliverables.

Topics

The basic content unit in DITA is the *topic*. According to the public review draft of the DITA 1.2 specification, "a DITA topic is a titled unit of information that can be understood in isolation and used in multiple contexts. It should be short enough to address a single subject or answer a single question but long enough to make sense on its own and be authored as a self-contained unit."

DITA topics follow a defined structure. This drives consistency across topics, especially for topics developed by different people. For example, a task topic might include, in order:

- a title,
- pre-requisites (what the learner needs to know, have, or do before performing the task),
- the task's steps,
- the expected result of performing the task, and
- post-requisites (what the learner needs to do after completing the task).

You can have DITA enforce as much of that structure as you want. For example, you might require every task to have a title, background information, steps, and expected results, but leave the inclusion of pre-requisites and post-requisites to each topic's individual developer.

DITA 1.0 and 1.1 by default provide three basic topic types, each with its own default structure and enforcement. You can use *concept topics* for overview and high-level explanatory information, *task topics* for information about performing specific tasks, and *reference topics* for detailed information such as system specifications, part numbers, and command syntaxes. DITA 1.0 and 1.1 also provide a generic topic type that DITA developers can specialize to their needs. DITA 1.2 introduces additional default topic types, some of which are part of the L&TC Specialization. I will describe these later under "What is the DITA L&TC Specialization?"

By this point, you might be thinking, "Whoa, this DITA stuff sounds pretty complex." Admittedly, there *are* some highly complex aspects of DITA, but the basic mechanics of developing a topic in DITA are relatively simple. Figure 1 is the DITA code for a simple task topic.

```

<task id="MakingTomatoSandwich">
  <title>Making a tomato sandwich</title>
  <taskbody>
    <prereq>To make a tomato sandwich, you will need:
      <ul>
        <li>a sharp knife</li>
        <li>a cutting board</li>
        <li>one or more fresh tomatoes</li>
        <li>two slices of bread</li>
        <li>mayonnaise</li>
      </ul>
      <image href="images/sandwich.jpg"/>
    </prereq>
    <steps>
      <step><cmd>Using the knife, spread the mayonnaise on one side of each slice of bread.</cmd></step>
      <step><cmd>Place the slices of bread, mayonnaise side up and side by side, on one end of the cutting board.</cmd></step>
      <step><cmd>Using the knife and the other end of the cutting board, cut the tomato or tomatoes into 1/4" slices.</cmd></step>
      <step><cmd>Place the tomato slices on one slice of bread. </cmd></step>
      <step><cmd>Place the other slice of bread, mayonnaise side down, on top of the tomato slices.</cmd></step>
    </steps>
    <result>You now have a tasty tomato sandwich!</result>
    <postreq>Eat the sandwich before the bread gets soggy.</postreq>
  </taskbody>
</task>

```

Figure 1: Sample DITA code for a task topic

If you're familiar with HTML code, this should look somewhat familiar. Element tags are enclosed in "<" and ">" brackets, a "/" indicates a closing tag, and the "" and "" tags indicate an unordered list of the task pre-requisites. If you're not familiar with HTML code, though, rest assured there are many editors available that will help you develop DITA content in a WYSIWYG environment.


Figure 2 shows how the same task might look in a DITA editor.

Making a tomato sandwich

Pre-requisites:

▼ To make a tomato sandwich, you will need:

- a sharp knife
- a cutting board
- one or more fresh tomatoes
- two slices of bread
- mayonnaise



Step 1
Using the knife, spread the mayonnaise on one side of each slice of bread.

Step 2
Place the slices of bread, mayonnaise side up and side by side, on one end of the cutting board.

Step 3
Using the knife and the other end of the cutting board, cut the tomato or tomatoes into 1/4" slices.

Step 4
Place the tomato slices on one slice of bread.

Step 5
Place the other slice of bread, mayonnaise side down, on top of the tomato slices.

Result:

▼ You now have a tasty tomato sandwich!

Post-requisites:

▼ Eat the sandwich before the bread gets soggy.

Figure 2: Developing a task in a DITA editor

Notice that Figure 2 shows some text not seen in the DITA code in Figure 1, such as "Pre-requisites." You can have your DITA editor automatically display text based on specific tags. In this case, when I told my editor I was writing the pre-requisite, the editor knew to precede that text with the heading "Pre-requisites." This not only saves typing time, but also helps drive consistency across topics. Subject to the features of your particular DITA editor, you can control what text appears automatically.

If your editor doesn't support the automatic display of text, you can still have text automatically appear in your final deliverables. I'll explain that more under "Output Formats."

Let me point out a couple more things in these figures.

First, in Figure 1, notice how the tags describe the nature of content they contain and not the formatting. For example, "<title>" indicates the topic's title and "<step>" indicates a step. We refer to this as *semantic tagging*. With semantic tagging, you indicate the nature of the content and leave the formatting for a later stage. Figure 2 shows some text as boldfaced or larger, but this is purely for the convenience of the developer; how the text will look in your final deliverable might be different. Again, more about this under "Output Formats."

Second, while the two figures include an image, you can include just about any type of multimedia content in a DITA topic: audio, Flash movies, and so on. Like HTML, you can also provide hyperlinks to content on the Internet. Your main practical consideration is the format of your final deliverable. (From my experience, animated GIFs don't animate very well in hardcopy!)

Maps

As I explained earlier, topics are the basic content units in DITA. A set of DITA topics might include information about and for multiple products, audiences, and purposes. For example, you might develop topics about three products: the KAT-1000, KAT-2000, and KAT-3000. Many of your topics apply to all three products, but some apply only to one or two. Many of your topics apply to all audiences, but some are just for experts. Many of your topics support both training and product documentation, but some support only one or the other.

To create a specific deliverable for a specific product, audience, and purpose, you use a DITA *map*. A DITA map is essentially a list of pointers to a set of specific topics, specifying what topics to include in the deliverable. A map also specifies the order and hierarchy of the topics, and provides navigation, such as a table of contents and cross-topic links in the final deliverable.

You can create maps anytime in the DITA process. Some DITA developers use maps as a planning and tracking tool before topic development, while others create maps after they develop the topics. In any case, you need complete maps and topics to create complete deliverables.

Figure 3 illustrates how maps can create multiple deliverables from a single set of DITA topics. One deliverable is a KAT-2000 training course for beginners, and the other is an administrator guide for the KAT-3000. Notice how each map includes two of the topics, "How to Print" and "Error Messages," but neither map includes "About the KAT-1000." Only one map or the other includes the remaining topics. The maps also specify the order of the topics, and if the topics were hierarchical, the maps could specify that as well.

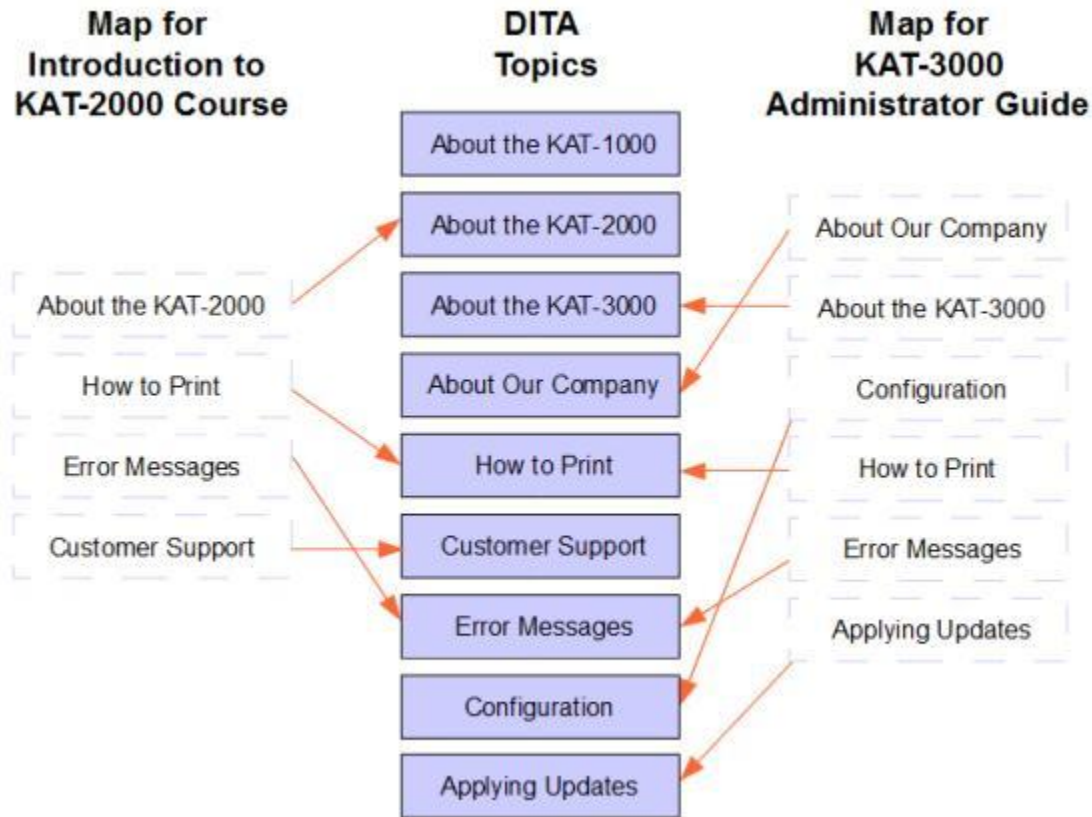


Figure 3: Assembling DITA topics into maps

In a real DITA environment, you should expect a much higher percentage of multiple-map topics than is shown in Figure 3. It's common to use 50% or more of your topics in multiple maps.

Note that topic content doesn't physically exist in maps. Maps only contain pointers (known in DITA as *content references*) to topics. Consequently, if a topic changes, any map pointing to that topic will reflect that change. This is generally a good thing because it enables DITA developers to make a change in one place and have that change reflected in multiple places at once. That said, you do need to be careful about changing shared content lest you effect a wrong change in another deliverable.

Recall that the DITA 1.2 specifications defined a DITA topic as "a titled unit of information that can be understood in isolation and used in multiple contexts." Maps help illustrate why it's important to have only context-neutral content in as many of your DITA topics as possible. For example, as soon as you use transitional text such as "Previously, you learned..." or "In Topic ABC..." , you set up a dependency between two topics. But if, in one deliverable, the other topic doesn't come before this topic or isn't included at all, you'd be pointing people in the wrong direction or to content they don't have. If you decide a specific deliverable really needs transitional text among topics, there are ways to do that, but that's beyond the scope of this article.

Despite what Figure 3 implies, including a topic in a map is not an all-or-nothing proposition. DITA topics and maps can make extensive use of *metadata* to support conditional processing (filtering), allowing customized topics during map processing. For example, let's say the three "About the..." topics contain exactly the same content except for the opening sentence. You could create a single "About the..." topic, write three different opening sentences in that topic, then use metadata to indicate which sentence

applies to which product. Corresponding metadata in the map can determine which sentence gets included.

Further, it's possible to use a single map for multiple deliverables. For example, if both the Introduction to KAT-1000 and Introduction to KAT-2000 courses contain the same set of lessons and exercises, but the content within those lessons differs slightly between products, one change in the DITA map could have the map include only the content that is common to both products and specific to one of the products.

Maps can point not only to topics, but also to other maps. You might use this feature, for example, to create a map for each lesson in a course, then use another map to determine which lessons to include in the course and in what order. As with topics, if something in a map changes, that change reflects in any maps that point to the changed map.

Admittedly, Figure 3 illustrates a very simple example; I've heard that IBM currently maintains more than 1,000,000 DITA topics, and who knows how many maps. Even so, the figure begins to illustrate the power and flexibility of DITA. And, as you'll now learn, output formats extend that flexibility even further.

Output formats

DITA output formats allow you to take the content specified by a DITA map and generate various deliverables suitable for printing, posting to the Web, e mailing, or sharing over mobile devices. By default, the DITA-OT provides output formats for XHTML, Compressed HTML Help (.chm), PDF, Eclipse Help, JavaHelp, Rich Text Format (RTF), and others. The DITA-OT notably does *not* support Microsoft PowerPoint, but I know of at least one commercial tool that does. Other DITA tools may offer additional output formats, and it's possible (although not necessarily easy) to develop your own output formats.

Because DITA uses semantic tagging instead of format tagging, output formats can completely control what your deliverables look like. Many DITA developers modify the default output formats to match the templates they developed in previous tools such as Microsoft Word or Adobe FrameMaker. Modifying PDF formatting can be especially tricky, but an increasing number of tools make this job easier.

Earlier, I explained that while your DITA editor might display and format text automatically when you're developing topics, it may not truly reflect what that topic will look like when you output it. Regardless of the editor, your selected output format controls the final look and feel of a deliverable. A DITA *publishing engine* uses the specifics of an output format to determine text-level formatting (typeface, size, color); if specific text or images are added to certain types of content (maybe a little pencil icon for notes); page layout (margins, the placement of headers and footers), and so on. Each output format includes the appropriate settings for the target file type. For example, logical page breaks can be important in PDFs, but are generally meaningless in HTML Help. For XHTML output, you might want to specify that external links open in a new window, but in a printed training manual, how to handle external links doesn't apply.

Sidebar: *What about Content Management Systems?*

People often discuss using Content Management Systems (CMSs) with DITA. CMSs provide a variety of useful features such as source control and review tracking for managing DITA topics, maps, and output formats. While many DITA implementations can benefit greatly from CMSs, DITA does not strictly require one. CMSs can be expensive to implement, so carefully analyze your specific needs.

Each time you want to generate a deliverable, you use a DITA publishing engine to process that deliverable's DITA map with an output format. Each DITA map can work with any supported output format, so if you want to publish a training manual in both PDF and HTML, you can use one DITA map and two output formats.

Consider now the extreme flexibility available for every DITA topic you develop. You can use a map to include the topic, specify the order and hierarchy of the topic, include only the topic content that's relevant

to a specified product, audience, and purpose, and generate the deliverable containing that topic using different output formats. If something changes in the topic, you simply regenerate the output.

What is the DITA L&TC Specialization? (AKA, Here's Why you Should Care about DITA)

In the opening to this article, I mentioned the DITA Learning and Training Content (L&TC) Specialization, and how IBM used it to automate content reuse, synchronize content between product documentation, and save on translation costs.

Specialization is a feature that provides additional flexibility for DITA developers. While the three basic topic types (concept, task, and reference) available in DITA 1.0 and 1.1 generally work well for technical publications, they don't always work so well for training deliverables. For a training deliverable, you might want to have topic types specialized for learning summaries and assessments, as well as element tags specialized for learning content.

Since the introduction of DITA, some training organizations have successfully developed and implemented their own DITA specializations. To create their specializations, the organizations used the inheritance features of DITA to design new topic types and elements from existing types and elements. Recognizing a widespread need for training specialization, OASIS formed the DITA L&TC Specialization subcommittee in 2006. Chaired by John Hunt of IBM, the subcommittee had participation from major IT companies, the U.S. Department of Defense, educational publishers, tool vendors, and consultants.

The result was the DITA L&TC Specialization. According to the public review draft of the DITA 1.2 specification, the subcommittee designed the Specialization to:

- "Provide a general top-level design for authoring of education content with good learning architecture, following DITA principles and best practices,
- Establish guidelines that promote best practices for applying standard DITA approaches to learning content,
- Provide basic support for processing DITA content for delivery as learning and training in a variety of forms, including print and presentation delivery to support instructor-led training (ILT) and web delivery for distance learning,
- Provide a framework for developing targeted support for processing DITA learning content for delivery with standards-based learning, specifically targeting SCORM, and
- Build on existing DITA infrastructures as much as possible, so learning content developers do not have to start from scratch."

While the subcommittee based the Specialization on training industry best practices, the Specialization does not focus on any specific instructional model.

Although the DITA 1.2 specification is not yet formally approved, many DITA tool vendors have already implemented the specification, including the L&TC Specialization. You can use it today.

The L&TC Specialization provides a wealth of features for instructional developers. It includes topic types specialized for learning plans, overviews, content, summaries, and assessments, and numerous specialized element tags such as those for instructor notes and for assessment interactions (open question, true/false, single select, and so on). The Specialization also supports learning-specific metadata such as typical learning time and difficulty level.

Due to the nature of DITA inheritance, even if you or other developers create DITA topics outside the L&TC specialization, you can still use L&TC and non-L&TC topics together. For example, you might want to create a lesson map that starts with an L&TC overview, then reference in several non-L&TC topics from the product documentation, then finish with an L&TC summary and assessment. The DITA basic concepts I've described in this article all apply to the L&TC Specialization, so for example, you would build lessons or entire courses using DITA maps, and generate final deliverables using output formats.

Finally, notice that the Specialization targets the Sharable Content Object Reference Model (SCORM). In addition to the other default DITA output formats you can use the L&TC Specification to output SCORM 2004 e-Learning packages, then import those package into a SCORM-compliant Learning Management System (LMS). A discussion of SCORM is beyond the scope of this article, but plenty of other *Learning Solutions Magazine* articles cover SCORM in detail.

So why should you care about DITA and the L&TC Specialization? Let's say you work with large amounts of instructional context, copy and paste content from other sources, or work with large teams and need better content consistency. Then DITA and the L&TC Specialization may be your best way to address the dilemma of learners having less time and money to learn what they need to know, and you having less time and money to deliver what those learners need. If the L&TC Specialization drives DITA 1.2 adoption in the training development world as quickly as DITA 1.0 and 1.1 drove adoption in the technical documentation world, you can probably expect your manager to start asking soon if DITA is right for your organization.

Summary

Since DITA's introduction, many organizations have realized significant time and cost savings, as well as increased quality and flexibility, by using DITA for technical documentation. They have seen other benefits from DITA as well, including more collaborative development environments.

With the introduction of the L&TC Specialization, DITA's benefits are now more easily accessible to instructional developers, and there is greater opportunity for collaboration between training and documentation teams. I realize that many training and documentation teams operate in separate silos, and that technology alone won't address every issue of politics and varying approaches to content, but the business case for DITA can be very compelling. Is implementing DITA easy? No. Does it pay off in the long run? If carefully planned and executed, probably.

This article provides a whirlwind tour of DITA and the L&TC Specialization, but I hope it provides enough information for you to decide if DITA is worth exploring further. I would love to hear your feedback.

References and resources

Hunt, John, "IBM leverages the DITA Learning and Training Specialization" presentation to the RTP Boston DITA Users' Group, June 23, 2010.

OASIS, Public review draft of the Darwin Information Typing Architecture Version 1.2 specifications, <http://lists.oasis-open.org/archives/dita/201007/msg00004.html>

OASIS Website, <http://www.oasis-open.org> and <http://dita.xml.org>

SourceForge, The DITA Open Toolkit Project Home page, <http://dita-ot.sourceforge.net/>

Learning Solutions Magazine, ©2010 eLearning Guild