



WHITE PAPER

Planning for DITA Success Part Two: How to Deploy DITA, Step-By-Step

By Steve Manning of The Rockley Group and Su-Laine Yeo and Paul Prescod of XMetaL



Steve Manning,
The Rockley Group



Su-Laine Yeo,
XMetaL



Paul Prescod,
XMetaL



DITA, by helping organizations migrate to and leverage XML, is revolutionizing the way modern document publishing teams author, manage, and deliver information. Part One of this series, *'Planning for DITA Success: How to Set Up the Right Team and the Right Strategy'*, outlines pre-deployment considerations for success, including how to line up the right strategy and people.¹

This paper, the second of the series, closes the loop by examining implementation issues from a technical perspective. It explores best practices within the five key steps of a successful transition:

1. Selecting DITA Solution Components
2. Choosing to Specialize DITA (Or Not)
3. Converting Legacy Content
4. Configuring Tools
5. Launching DITA and Creating Content

¹ To download Part One, please visit www.rockley.com or www.xmetal.com.



“Because DITA is a standard rather than a specific product, organizations have more options and greater responsibility in implementing it.”

Because DITA is a standard rather than a specific product, organizations have more options and greater responsibility in implementing it. Documentation teams face important decisions: should they specialize, or not? What authoring software should they use? Should they use a content management system? If so, how will they select it? What sort of publishing technologies are applicable? All these considerations affect DITA's results, yet many organizations do not systematically address them prior to making the transition.

Before any DITA endeavor, even a piecemeal one, address these issues at a high-level. Consider how your organization's people, processes and technology interact in the endeavor of authoring and publishing. Which methods are effective? Which are a liability? The transition to DITA can be a fresh start, a chance to identify and eliminate bad habits and inefficiencies. Consider this paper a guide to making these valuable assessments, and beginning the journey to DITA on the right path.

STEP ONE: SELECTING DITA SOLUTION COMPONENTS

First, DITA publishing requires DITA-formatted content. This requires the conversion of documents from other formats, or direct authoring within DITA. Most organizations will do a mix, converting pre-existing content once and then maintaining it in DITA thereafter.

The maintenance process will require some kind of **DITA-aware authoring application**. You will also need a **publishing engine** to generate a variety of output formats, such as PDF, Microsoft Windows Help, HTML for the Web or others. Beyond these two required components, you may also benefit from a subset of the following optional ones:

Content Management Most organizations adopt a content management system (CMS) during the transition to DITA. At a minimum, a CMS allows for easier storage, versioning and searching of content components, plus additional features such as translation and workflow management.

Translation Management To publish to multiple languages, you will need some degree of translation management functionality to manage the relationship between translated and original components, as well as to automatically reuse translated content.

Workflow Management A workflow management system automates the routing of information from one person to another. Rather than manually sending ‘Please review/translate/approve this content’ emails, the workflow system notifies contributors with a consistent, pre-defined workflow for each topic and map.

To select all relevant solution components, examine the issue of content contribution: how will subject matter experts outside the documentation team (engineering, legal, marketing, customer service) assist in creating and validating



“If the authoring tool is easy-to-use, end users and contributors will adopt the DITA system enthusiastically.”

documentation? Make sure, if required, that your chosen components allow domain contributors to create original content (e.g. system requirements for a software product, or electrical specifications for hardware) and review proposed content to ensure accuracy.

DITA Authoring Software

The XML foundation of DITA requires XML authoring software. If the authoring tool is easy-to-use, end users and contributors will adopt the DITA system enthusiastically. A short learning curve minimizes training costs, accelerates productivity and generates quick wins beyond the core documentation team. Since DITA is a form of XML, even a basic tool such as Notepad can create DITA topics. But a full-featured, DITA-oriented XML authoring application is more robust and production-worthy. In order to ensure both high usability and the right functionality fit, consider the following:

How seamless is the experience of writing within the application?

Writers should focus on creating content rather than be distracted by the tactical mechanics of XML. Choose an authoring application with topic and element templates that reduce the need to memorize the structure of DITA topics.

Does the application support special elements and attributes such as cross-references, content references and conditional text?

Look for attributes that are editable either by typing or selecting values from lists. Make sure the tool easily aligns to your own metadata values and enables the visualization of reused content such as content references and conditional text.

Does the application create valid DITA with the meanings you intend?

Applications should create DITA content that not only passes validation, but also works the way you expect. For example, DITA conditional text requires that spaces be used between multiple attribute values. In some tools, features for conditional text may use non DITA-compliant delimiters such as semicolons.

How does the application support editing DITA maps?

Rather than manually editing the map XML, choose a tool that allows visual map manipulation with a graphical user interface. It should illustrate the hierarchical map structure as a tree (similar to a table of contents), and allow for easy reorganization of the tree.

Does the application make it easy to create deliverables?

Instead of having to download, install and configure a separate publishing engine, look for an application that integrates publishing and previewing into the authoring application. You may even be able to make these functions accessible via simple menu selections instead of requiring authors to use a command line or separate application.

Does the application have rich word processing features?

The editor you choose should include all the functionality authors need—not just enough functionality to add markup. This includes support for word processing



“Because DITA is standards-based, DITA can publish content using a variety of different techniques—through an XSL-FO stylesheet to PDF, an EDD to FrameMaker, an XSL stylesheet to HTML or the Microsoft Help Workshop to Windows HTML Help.”

tasks such as spell check, character-level revision marks and graphics insertion and visualization. These essential features are often lacking in immature XML authoring applications.

Publishing DITA

Because it is standards-based, DITA can publish content using a variety of different techniques—authors can publish through an XSL-FO stylesheet to PDF, an EDD to FrameMaker, an XSL stylesheet to HTML or the Microsoft Help Workshop to Windows HTML Help. But to do so, DITA content must be assembled from its maps and topics with content references resolved and conditional text filtered out. This requires a component that will interpret DITA's advanced features and simplify them for DITA unaware output tools. The most popular way of achieving this is the DITA Open Toolkit, an open-source project based upon work done within IBM.

Configuring the DITA Open Toolkit to run on your system can be challenging, since it requires different XML processors for transformations and conversions. Having ANT—a scripting/build environment that helps to create and run projects—can make configuration easier. However, you either need the ability to set up and configure Java applications on your system, or access to someone who does—which can be daunting for many technical writers. Some solutions have an automatically configured Toolkit built-in so that output can be created using simple menu commands. If all goes well, publishing DITA simply requires the push of a button. If you have very specific requirements (e.g. publishing to handheld computers) you may need to dig into the DITA Open Toolkit.

Content Management

While DITA itself does not impose any specific requirements for a content management system (CMS), investing in one enhances DITA's benefits. DITA can be used in conjunction with a traditional file management system as a content repository, but a CMS is more practical and production-worthy. The key determining factor is volume—the more content you produce, the more sense it makes to include a CMS.

In the past, traditional tools such as Microsoft® Word and Adobe® FrameMaker® stored content in files, with each file representing a chapter or section. With DITA, content is saved as topics, typically in separate files. With multiple topics in a single chapter, the number of individual files associated with a book has the potential to grow significantly. This may strain the capabilities of file-system-based content management, no matter how complex the naming scheme used.

Authoring has not traditionally featured modular reuse. If an author wanted to include sections from another information product, these were usually copied and pasted into the new document, resulting in a complicated and error-prone flurry of updating procedures. DITA emphasizes reuse (‘write once, use many’), meaning that changes to a single source file are automatically disseminated to all different outputs in which it appears.



A shift to DITA prompts organizations to analyze how they can most effectively create, reuse and leverage content. Folders and naming schemes may be insufficient to categorize topics—and if you can't find it, you can't reuse it. A CMS, simply by virtue of its ability to identify and reuse topics and components, may be worthwhile. Additionally, a CMS with element-level searching helps in the identification and reuse (through *conref*) of existing content. When evaluating a CMS, consider the following:

- How powerful are its search capabilities, including full-text search?
- Does it embed workflow capabilities or can it be integrated with external workflow engines?
- Does it have native XML features or does it treat XML as just another data type?
- Does it have DITA-specific features, such as the grouping of topics based upon their relationships within a map?
- Do link management features determine whether a change in a topic or referenced element in one context will have an impact on multiple deliverables?
- How extensible is the CMS? Can you influence its actions when content is checked in and checked out? Can you extend its user interface?
- Is the CMS tightly integrated into your authoring application of choice?

Content management systems vary widely in their features, goals and paradigms. Evaluate them carefully and methodically to make sure they will deliver the right fit and value for your organization.

Workflow Management

A workflow management system routes information and tasks from one individual (or role) to another. For example, an organization might define a workflow that dictates documents to be initially defined by an information architect, with topics then assigned to various authors. Each topic would have its own workflow involving writing, reviewing and approval. Then, topics are merged back into the main deliverable for overall review and approval.

Workflow systems ensure that each individual is assigned the right task at the right time, and that the content flows smoothly through creation, approval and publishing. It should present a relatively simple mechanism for setting unique workflow rules—for instance, your organization might have a policy that every topic change be reviewed by a subject matter expert. If the review fails, the topic is routed back to the author.



“If the process is carefully managed, the reuse of content through DITA has the potential to dramatically reduce translation costs.”

Translation Management

The reuse of content through DITA has the potential to dramatically reduce translation costs, but only if the process is carefully managed. A variety of tools, technologies and techniques can be applied to do this. When evaluating your options, address the following basic considerations:

1. How will content be exchanged between authors and translators?
Do translators use the same content repository (e.g. CMS), or do sets of topics have to be bundled and sent to them? If the latter applies, can this process be automated? If some topics have not changed, can they be marked to avoid redundancy of effort?
2. How will content be passed off to translators? Does the workflow system manage the flow of content from authors to translators? If not, do you have any kind of translation-specific workflow system that can do so?
3. When your content arrives at the translators, do they have tools to accelerate turnaround? In particular, a translation memory system can recognize phrases that have already been translated and suggest them automatically, on the basis that the same phrase is often (but not always!) translated the same way regardless of context.

Content Contribution Tools

On most teams, subject matter experts occasionally contribute content to DITA topics. These users can use existing tools such as Microsoft Word, or author directly in DITA. If they are likely to copy and paste from a word processor into DITA, test this when evaluating DITA authoring tools. Tools vary greatly in how (if at all) they translate Word content to DITA.

If contributors use DITA directly, content will require less cleanup and be more easily updated. For these users, web-based DITA authoring interfaces are a popular option. These typically have a simplified feature set with lower licensing and deployment costs.

Many document owners solicit feedback by distributing paper or PDF copies, which is labor-intensive and error-prone. An XML-based reviewing process is much more efficient, improving the quality, manageability and trackability of input. For example, the web-based XMetaL Reviewer enables multiple reviewers to comment at the same time and to see each others' comments. A web-based reviewing environment allows contributors to be productive quickly without having to install complex and potentially confusing software. XML-based reviewing also allows easy round-tripping of comments across drafts.

STEP TWO: CHOOSING TO SPECIALIZE DITA (OR NOT)

At the earliest stages of DITA's evolution, its developers recognized that many organizations had topics that would not fall into pre-defined categories such as Concepts, Tasks or References. DITA DTDs and stylesheets are designed to



facilitate the creation of unique topic and element types, or the modification of existing topic and element types to meet specific needs. ‘DITA specialization’ refers to the amending of defined structures, or the addition of new topic types or elements to supplement structures provided in base DITA (or another specialization).

The DITA 1.0 standard is restricted to the specialization of elements but not of attributes. However, most organizations add their own custom attributes in anticipation of the DITA 1.1 standard which will formally include support for such customization in mid-2006. Organizations customize the DITA DTD or schema by taking one of the following actions: adding new topic types; adding new elements to existing topic types (i.e. semantic elements replacing generic elements); revising existing topic types or attributes; or adding attributes.

Depending on your organization’s needs, your application of DITA will fall into one of three degrees of specialization, as follows:

1. For simply-structured information, the four built-in DITA topic types (topic, concept, task, and reference), five domains (programming, typographic, software, user interface and utilities) and one map type may be sufficient. This option is not tailored to unique needs, but presents the easiest and quickest deployment path.
2. For information chunks that are fairly common but are outside the basic set (i.e., those employed in e-learning or troubleshooting content), source additional specializations from within the DITA user community to amend the basic set.
3. Information that is uniquely structured requires unique specializations. Labor-intensive at the outset, this path results in a highly tailored system that will be relevant to internal users and optimized to produce exactly the results you require.

To Specialize Or Not?

“Using built-in DITA topic types is the most generic and cost-effective, while creating unique specializations is a highly tailored approach that requires a larger investment.”

Using built-in DITA topic types is the most generic and cost-effective approach, while creating unique specializations is a highly tailored approach that requires a larger investment. If economical, generic solutions suffice, then of course you should do so. But how will you decide? When evaluating the need to specialize, consider the following criteria:

Is tight structural control required?

Broad authoring groups often benefit from tight control of structural elements. Element names should provide specific indications of what content should be entered, and you may also need to replace some of the more generic elements in DITA with specific semantic elements. A specialized schema or DTD can help both the organization and the writer by enforcing consistency, and by guiding writers along rules designed for consistency, clarity and searchability.



“The DITA architecture is designed to support reuse, and does not require specialization to do so.”

Will you reuse content in a complex-and-granular or simple-and-coarse-grained manner?

The DITA architecture is designed to support reuse, and does not require specialization to do so. However, specializing can make reuse easier to manage. For instance, if you are using DITA to write ‘Troubleshooting’ and ‘How-To’ articles for a knowledge base, it might be easier to find these topics if you create corresponding topic types.

How complex are your output requirements?

If you can’t find it, you can’t manipulate it. In order to filter content and change formatting for different outputs or uses, clearly identify it with a semantic element name or attributes. This way, the stylesheet can apply the required manipulation to the correct element. For example, base DITA has an element for unordered (bulleted) lists. Consider specializing this element into a “checklist” that will be rendered with checkboxes. In an online context, the checklist could be clickable to help readers progress through it.

If you only require loose control with simple reuse and output, generic, out-of-box base DITA will be sufficient—even large organizations have chosen this route. However if you require tight control with complicated reuse models and highly customized output (especially print output), you may need to specialize.

Depending on your industry, an appropriate specialization may already exist. Refer to the DITA focus area maintained by OASIS for the latest on leverageable resources. An inquiry on a DITA mailing list may also turn up someone who is in the midst of developing one that meets your needs.

Some forms of authoring control can be accomplished without specialization. Base DITA includes elements and attributes that are very similar and not required in your content, such as unordered lists and simple lists. Rather than removing these extra elements and attributes from the DTD or schema, simplify the working environment by hiding them. Optimize your authoring environment for the subset of DITA markup that you need, without unwanted elements confusing the workspace. (See “1.5.1. Configure the Authoring Interface”).

Best Practices for Specialization

If you decide to specialize:

- Understand your content. Specialization must fit to content—not as it is today, but as you would like it to be structured after the move to topic orientation.
- Have clear motivations. There are many reasons for specializing, all of which need to be clearly articulated to make effective design choices.
- Build on success. Find organizations that have already specialized in a similar way, and gain ground by collaborating or leveraging their work. At the least, use their specializations as a starting point. Either way, you will learn from their decisions and experience.



- Extend your DITA knowledge as much as possible. Many aspects of working with DITA are not apparent without some depth of experience. Designate at least one person from your team to become a DITA expert to ensure your specialization aligns with proven architectural principles.

STEP THREE: CONVERTING LEGACY CONTENT

Content conversion is one of the more subtle issues in the DITA transition process. Many tools are available for converting from legacy formats to XML, and they work by mapping styles to XML elements. The right technique will depend upon a variety of factors:

How much content do you need to convert?

Greater volumes of content require automation of the conversion process. There is no need to convert everything your department has ever written—instead, focus on documents with the most strategic value going forward. Some companies leave legacy content in older formats and use DITA to develop new content.

How consistent is the input format?

Some organizations have strict style definitions that are applied in applications such as Word or FrameMaker, while others' styles are inconsistent and subject to the preferences of individuals. In the latter case, automated conversion to DITA will be more challenging. One organization may have a style for 'task step'; another may use a generic numbered list; and a third may modify paragraphs by typing the actual numbers. In the first case, conversion directly to DITA task topics may be possible, with the task step style mapping to the equivalent DITA element type. In the second case, automated conversion to a numbered list in a more generic DITA topic type is possible, but human intervention may be needed to massage the content from a generic topic into a task. In the third case, the automated conversion would not even recognize the list as a list. This will add extra effort in the marking up of task steps.

Is the current content already topic-oriented?

If the current content is written in a book-oriented fashion with dependencies between sections, content conversion may not be sufficient. Some rewriting will be necessary to adapt the structure, either before or after the content is converted. Some conversion tools can create individual topic files automatically, based on the heading structure of your document.

How much duplication and/or reuse is there in your legacy content?

We generally recommend that you do not convert redundant content twice. Implement your content reuse strategy in parallel with the conversion process. If legacy content includes conditional text, cross-references, variables, text insets or index markers, translate these to DITA equivalents. A DITA-optimized authoring application can make it easy to manually implement the equivalent DITA markup for these features once content is in XML. In general, ensure that all conditional text is displayed before you convert the content.

“A DITA-optimized authoring application can make it easy to manually implement the equivalent DITA markup once content is in XML.”



How rich do you want the converted documents to be?

To save time, consider converting legacy content to generic DITA topics instead of Concept, Reference, and Task topic types. Use the more specific topic types for new content.

Large volumes of consistent, topic-oriented content tend to require a highly automated conversion procedure that can run over thousands of files without stopping. A broad variety of tools exist to help with that degree of batch conversion. Smaller volumes of content can be converted with tools built into your outgoing or incoming authoring applications. For example, if converting from Word to DITA using XMetaL Author DITA Edition, use the cut and paste feature to generate DITA. Or if converting from FrameMaker to DITA, use FrameMaker's conversion tables feature.

Consider doing a small conversion pilot project to test the process and to understand the tool's capabilities. When testing, check that converted content (particularly graphics) appears as it should in all output formats. When the process is validated, a temporary worker could assist with the more systematic aspects of conversion.

STEP FOUR: CONFIGURING TOOLS

One of the main attractions of DITA is the opportunity to either use tools and stylesheets out-of-the-box, or tailor to each team's requirements. Traditional authoring systems require the setup of styles, templates, macros, and conditional text settings, and setting up a DITA-based system is similar.

Configure the Authoring Interface

Configure the DITA authoring interface with a mind for improving usability, shortening the learning curve and maximizing efficiency. As you would use a non-DITA tool to set up styles or conditional text tags, configure a DITA tool for unique elements and attributes. The ease of configuring an authoring application varies, and can be quite an undertaking. But changing the interface is much safer than changing the DITA DTD or schema. Changing the interface will not affect the displaying of DITA documents created with other tools, nor will it require the changing of DITA stylesheets for output.

Begin by setting up conditional text with your required attributes and values. Choose DITA metadata elements and attributes, and decide whether to store this information in topic or map files. If your authoring application provides dialog boxes to edit DITA metadata elements and/or attributes, add or remove fields so that the dialogs include only the fields required.

If authors will repeatedly enter the same metadata values, create checkboxes, radio buttons and drop-down lists, and configure controls to default to the most common or safest value.



Add menu items or dialogs to insert specialized elements, and configure the authoring stylesheet(s) to display the new elements. Hide or remove commands for unused base DITA elements. Some DITA elements are grouped into domains, and the authoring application may be able to hide entire domains. Finally, modify topic templates to include commonly used elements.

Customizing Output Stylesheets

Most organizations will customize DITA Open Toolkit stylesheets to accommodate corporate branding and formatting. The correct way to do this depends upon your publishing strategy and the details of the changes. For instance, simple Web output font changes can often be applied by changing a few lines of the CSS. On the other hand, radically changing the print formatting of a DITA table type might require some XSL-FO programming (for those using XSL-FO for print output) or FrameMaker EDD changes (for those using FrameMaker as an output engine). A need for specialized element formatting is another valid reason to customize the stylesheets.

Configuring Other Tools

The configuring of CMS, workflow systems, translation memories or other tools will vary. Get a clear picture from vendors as to how much configuration or integration is normal, and how intensive this process may be.

Checklist: Setting up a DITA Solution

In summary, following are the key milestones in setting up a DITA solution:

- Evaluate and select a DITA authoring application.
- Download and install the DITA Open Toolkit, or install an authoring application that includes the Toolkit. Select any other publishing technologies (such as FrameMaker for print) that you require for output.
- If necessary, add custom elements and attributes to DITA through specialization.
- Plan for the conversion of legacy content.
- Choose a method and strategy for storing and retrieving content, especially reused content (e.g. a CMS, or file system).
- Configure the authoring application's user interface to fit unique needs.
- Customize DITA stylesheets for each required output format.

STEP FIVE: LAUNCHING DITA AND CREATING CONTENT

A conventional documentation project would typically require working in one file for the entire deliverable, or perhaps one file per chapter. With DITA, authors work with one file for each topic, all of which point to the DITA map. Single-topic files make it easier to share a project among multiple authors, as each writer can own a subset of the project's topic files.



Create a map file, scoping all topics at the beginning of a project. The early creation of all associated files makes planning and output testing throughout the project much easier. Create a placeholder file for each topic, and distribute the placeholder files to authors to complete (a placeholder for a topic consists of a title and description or outline of the eventual content of the topic). Following is an example of a typical project-starting process.

1. Create a draft table of contents (TOC) of your document, and have it reviewed. Plan its navigational structure for any online deliverables. Draft the TOC using pencil and paper or a simple text editor.
2. When the TOC is stable, identify which existing topics can be reused and which new topics need to be developed. Create a placeholder file for each new topic.
3. Create a DITA map reflecting the table of contents, with references to the reused topics and to the placeholder files for the new topics. If you are using XMetaL Author DITA Edition, create a topic file and link to it from the map in one step.
4. If each author is working on multiple topics, set up a separate map file for each author. In addition, create a parent map file with links to each of the child map files.
5. Add related links to your topics, and/or set up the map file to generate related links and browse sequences. If multiple authors are working on the project, decide who will be the initial owner of each topic and distribute the placeholder files to authors accordingly.
6. Designate an owner of the DITA map file, and give each author a read-only copy of the map.

Find out more about how the XMetaL family of software applications can bring DITA to life within your organization. To watch a demo, download trial software or learn more, visit www.xmetal.com or call to speak with an account representative.

www.xmetal.com

North America (toll free) +1.866.793.1542

Europe +31.20.330.20.14

International +1.604.697.8701

Email products@xmetal.com

CONCLUSION

The right tools and technical strategy make a huge impact on the success of any DITA deployment. DITA and its associated tools can be easily implemented in a basic form, and adapted to unique needs over time. In this Part Two 'How to Deploy DITA, Step-By-Step', we've reviewed how to choose, customize and integrate the specific tools and technology that will power a new content model that fits your organization.

About XMetaL

XMetaL® is the world's leading XML content creation and collaboration solution, advancing the goals of globally-focused, customer-driven organizations. XMetaL's family of software and services enables your company to create, share, reuse, and deliver content more quickly and cost-effectively, across all customer channels, from sales, marketing, and customer service, to training and technical support. Visit www.xmetal.com to learn how leading brands such as Cisco, Daimler Chrysler, HP, Kodak, Microsoft, Philips, RIM, Southwest Airlines, Sybase, Symantec, Texas Instruments and USA Today use XMetaL to bring products and services to market faster, with greater accuracy, and at lower cost.

© 2006 XMetaL. All rights reserved.

Rev 04/06