

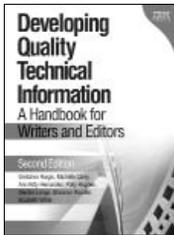
# DITA Best Practices

A Roadmap for Writing, Editing,  
and Architecting in DITA

Laura Bellamy, Michelle Carey, Jenifer Schlotfeldt



# Related Books of Interest



## Developing Quality Technical Information, 2e

By Gretchen Hargis, Michelle Carey, Ann Kitty Hernandez, Polly Hughes, Deirdre Longo, Shannon Rouiller, and Elizabeth Wade  
ISBN: 0-13-47749-8

Direct from IBM's own documentation experts, this is the definitive guide to developing outstanding technical documentation—for the Web and for print. Using extensive before-and-after examples, illustrations, and checklists, the authors show exactly how to create documentation that's easy to find, understand, and use. This edition includes extensive new coverage of topic-based information, simplifying search and retrievability, internationalization, visual effectiveness, and much more.



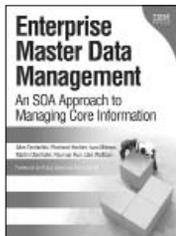
## The IBM Style Guide

By Francis DeRespinis, Peter Hayward, Jana Jenkins, Amy Laird, Leslie McDonald, and Eric Radzinski  
ISBN: 0-13-210130-0

*The IBM Style Guide* distills IBM wisdom for developing superior content: Information that is consistent, clear, concise, consumable, reusable, and easy to translate. Written by a team of senior IBM editors, this book helps any organization improve and standardize content across authors, delivery mechanisms, and geographic locations. *The IBM Style Guide* can help any organization or individual create and manage content more effectively.

Sign up for the monthly IBM Press newsletter at  
[ibmpressbooks/newsletters](http://ibmpressbooks/newsletters)

# Related Books of Interest



## **Enterprise Master Data Management** An SOA Approach to Managing Core Information

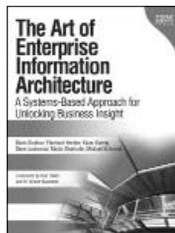
By Allen Dreibelbis, Eberhard Hechler, Ivan Milman, Martin Oberhofer, Paul Van Run, and Dan Wolfson

ISBN: 0-13-236625-8

The Only Complete Technical Primer for MDM Planners, Architects, and Implementers

*Enterprise Master Data Management* provides an authoritative, vendor-independent MDM technical reference for practitioners: architects, technical analysts, consultants, solution designers, and senior IT decision makers. Written by the IBM® data management innovators who are pioneering MDM, this book systematically introduces MDM's key concepts and technical themes, explains its business case, and illuminates how it interrelates with and enables SOA.

Drawing on their experience with cutting-edge projects, the authors introduce MDM patterns, blueprints, solutions, and best practices published nowhere else—everything you need to establish a consistent, manageable set of master data, and use it for competitive advantage.



## **The Art of Enterprise Information Architecture** A Systems-Based Approach for Unlocking Business Insight

By Mario Godinez, Eberhard Hechler, Klaus Koenig, Steve Lockwood, Martin Oberhofer, and Michael Schroeck

ISBN: 0-13-703571-3

Architecture for the Intelligent Enterprise: Powerful New Ways to Maximize the Real-time Value of Information

In this book, a team of IBM's leading information management experts guide you on a journey that will take you from where you are today toward becoming an "Intelligent Enterprise."

Drawing on their extensive experience working with enterprise clients, the authors present a new, information-centric approach to architecture and powerful new models that will benefit any organization. Using these strategies and models, companies can systematically unlock the business value of information by delivering actionable, real-time information in context to enable better decision-making throughout the enterprise—from the "shop floor" to the "top floor."

# Related Books of Interest



## Search Engine Marketing, Inc

By Mike Moran and Bill Hunt

ISBN: 0-13-606868-5

The #1 Step-by-Step Guide to Search Marketing Success...Now Completely Updated with New Techniques, Tools, Best Practices, and Value-Packed Bonus DVD!

In this book, two world-class experts present today's best practices, step-by-step techniques, and hard-won tips for using search engine marketing to achieve your sales and marketing goals, whatever they are. Mike Moran and Bill Hunt thoroughly cover both the business and technical aspects of contemporary search engine marketing, walking beginners through all the basics while providing reliable, up-to-the-minute insights for experienced professionals.

Thoroughly updated to fully reflect today's latest search engine marketing opportunities, this book guides you through profiting from social media marketing, site search, advanced keyword tools, hybrid paid search auctions, and much more.



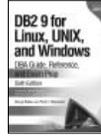
Listen to the author's podcast at:  
[ibmpressbooks.com/podcasts](http://ibmpressbooks.com/podcasts)



## Understanding DB2 9 Security

Bond, See, Wong, Chan

ISBN: 0-13-134590-7



## DB2 9 for Linux, UNIX, and Windows

DBA Guide, Reference, and Exam Prep, 6th Edition

Baklarz, Zikopoulos

ISBN: 0-13-185514-X



## Lotus Notes Developer's Toolbox

Elliott

ISBN: 0-13-221448-2



## IBM Lotus Connections 2.5

Planning and Implementing Social Software for Your Enterprise

Hardison, Byrd, Wood, Speed, Martin, Livingston, Moore, Kristiansen

ISBN: 0-13-700053-7



## Mining the Talk

Unlocking the Business Value in Unstructured Information

Spangler, Kreulen

ISBN: 0-13-233953-6

Sign up for the monthly IBM Press newsletter at  
[ibmpressbooks.com/newsletters](http://ibmpressbooks.com/newsletters)

*This page intentionally left blank*

# **DITA Best Practices**

# **DITA Best Practices**

**A Roadmap for Writing, Editing,  
and Architecting in DITA**

**Laura Bellamy, Michelle Carey,  
and Jenifer Schlotfeldt**

**IBM Press**

**Pearson plc**

**Upper Saddle River, NJ • Boston • Indianapolis • San Francisco  
New York • Toronto • Montreal • London • Munich • Paris • Madrid  
Cape Town • Sydney • Tokyo • Singapore • Mexico City**

**[ibmpressbooks.com](http://ibmpressbooks.com)**

The author and publisher have taken care in the preparation of this book, but make no expressed or implied warranty of any kind and assume no responsibility for errors or omissions. No liability is assumed for incidental or consequential damages in connection with or arising out of the use of the information or programs contained herein.

© Copyright 2012 by International Business Machines Corporation. All rights reserved.

Note to U.S. Government Users: Documentation related to restricted right. Use, duplication, or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract with IBM Corporation.

IBM Press Program Managers: Steven M. Stansel, Ellice Uffer

Cover design: IBM Corporation

Editor in Chief: Bernard Goodwin

Marketing Manager: Stephane Nakib

Publicist: Heather Fox

Acquisitions Editor: Bernard Goodwin

Managing Editor: Kristy Hart

Designer: Alan Clements

Senior Project Editor: Lori Lyons

Copy Editor: Apostrophe Editing Services

Proofreader: Williams Woods Publishing Services

Manufacturing Buyer: Dan Uhrig

Published by Pearson plc

Publishing as IBM Press

IBM Press offers excellent discounts on this book when ordered in quantity for bulk purchases or special sales, which may include electronic versions and/or custom covers and content particular to your business, training goals, marketing focus, and branding interests. For more information, please contact:

U.S. Corporate and Government Sales

1-800-382-3419

[corpsales@pearsontechgroup.com](mailto:corpsales@pearsontechgroup.com)

For sales outside the U.S., please contact:

International Sales

[international@pearson.com](mailto:international@pearson.com)

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both: IBM, IBM Press, Lotus and Notes.

UNIX is a registered trademark of The Open Group in the United States and other countries or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, SharePoint and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates

Adobe, the Adobe logo, FrameMaker, InDesign, PhotoShop and Illustrator are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Other company, product, or service names may be trademarks or service marks of others.

All rights reserved. This publication is protected by copyright, and permission must be obtained from the publisher prior to any prohibited reproduction, storage in a retrieval system, or transmission in any form or by any means, electronic, mechanical, photocopying, recording, or likewise. For information regarding permissions, write to:

Pearson Education, Inc  
Rights and Contracts Department  
501 Boylston Street, Suite 900  
Boston, MA 02116  
Fax (617) 671-3447

ISBN-13: 978-0-13-248052-9

ISBN-10: 0-13-248052-2

First printing August 2011

# Contents at a Glance

**Acknowledgments**

**About the Authors**

**Introduction**

## **PART I: WRITING IN DITA**

**Chapter 1**      **Topic-Based Writing in DITA**

**Chapter 2**      **Task Topics**

**Chapter 3**      **Concept Topics**

**Chapter 4**      **Reference Topics**

**Chapter 5**      **Short Descriptions**

## **PART II: ARCHITECTING CONTENT**

**Chapter 6**      **DITA Maps and Navigation**

**Chapter 7**      **Linking**

**Chapter 8**      **Metadata**

**Chapter 9**      **Conditional Processing**

**Chapter 10**     **Content Reuse**

## **PART III: CONVERTING AND EDITING**

**Chapter 11**     **Converting Content to DITA**

**Chapter 12**     **DITA Code Editing**

**Chapter 13**     **Content Editing**

---

# Contents

**Acknowledgments**

**About the Author**

**Introduction**

**Part I: Writing in DITA**

**Chapter 1 Topic-Based Writing in DITA**

- Books, Topics, and Webs of Information
- Advantages of Writing in Topics for Writing Teams
- DITA Topic Types
- Task Orientation
  - Task Analysis
- Minimalist Writing
- To Wrap Up
- Topic-Based Writing Checklist
- Task analysis form

**Chapter 2 Task Topics**

- Separate Task Information from Conceptual or Reference Information
  - Write One Procedure per Topic
  - Create Subtasks to Organize Long Procedures
- Task Components and DITA Elements
  - Titling the Task: <title>
  - Introducing the Task: <shortdesc>
  - Adding More Background Information: <context>
  - Describing Prerequisites: <prereq>
  - Writing the Procedure: <steps> and <steps-unordered>
  - Concluding the Task: <example>, <postreq>, and <result>
- To Wrap Up
- Task Topic Checklist

## Chapter 3 Concept Topics

Concept Components and DITA Elements

  Titling the Concept Topic: <title>

  Introducing the Concept Topic: <shortdesc>

  Writing the Concept: <conbody>

  Organizing the Concept: <section>

  Adding Lists: <ol>, <ul>, <sl>, and <dl>

  Including Graphics: <fig>, <title>, and <image>

  Highlighting New Terms: <term>

To Wrap Up

## Chapter 4 Reference Topics

Describe One Type of Reference Material per Topic

Organize Reference Information Effectively

Format Reference Information Consistently

Reference Components and DITA Elements

  Titling the Reference topic: <title>

  Introducing the Reference Information: <shortdesc>

  Organizing the Reference Information: <section>

  Creating Tables: <table>, <simplettable>, and <properties>

  Adding Lists: <ul> and <dl>

  Creating Syntax Diagrams: <refsyn> and <syntaxdiagram>

To Wrap Up

## Chapter 5 Short Descriptions

The <shortdesc> Element

  How the Short Description Is Used

Guidelines for Writing Effective Short Descriptions

  Briefly State the Purpose of the Topic

  Include a Short Description in Every Topic

  Use Complete, Grammatical Sentences

  Don't Introduce Lists, Figures, or Tables

  Keep Short Descriptions Short

Short Descriptions for Task, Concept, and Reference Topics

  Task Topic Short Descriptions

  Reference Topic Short Descriptions

Writing Short Descriptions for Converted Content

The <abstract> Element

  Using More DITA Elements in the Topic Introduction

  Including Multiple Short Descriptions

To Wrap Up

Short Description Examples

## **Part II: Architecting Content**

### **Chapter 6 DITA Maps and Navigation**

- DITA Map Structure
  - Include Topics in a DITA Map
  - Define Relationships Between Topics
- Information Organization
- Information Modeling
  - Benefits of Information Modeling
  - Building Information Models
- Bookmaps
- Submaps
- DITA Map Ownership
- Structure Content in a DITA Map
  - Group Topics Regardless of Hierarchy
- Reference Non-DITA Content
- Include Relationship Tables in DITA Maps
- Override Topic Titles and Short Descriptions
  - Navigation Titles
- Short Descriptions
- Suppressing Topics from the Table of Contents
- Suppressing Content from PDF Output
- Suppressing Content from HTML Output
- To Wrap Up
- Navigation and DITA Maps Checklist

### **Chapter 7 Linking**

- Hierarchical Links
- Inline Links
  - Link to Prerequisite and Postrequisite Information
  - Avoid Inline Links to Tables and Figures in a Topic
  - Create Inline Links to Repeated Steps
  - Create Inline Links to High-Level Tasks
- Control How Links Are Displayed
- Related Links
  - Relationship Tables
  - Related Link Element
- Collection Types
  - Sequence Collection Type
  - Choice Collection Type
  - Family Collection Type
  - Determining Which Collection Type to Use
  - Collection Types in Relationship Tables
- Links Created with the Importance Attribute

- Linking Scope
  - Local Links
  - External Links
  - Peer Links
- Link Testing
- To Wrap Up
- Linking Checklist

## **Chapter 8 Metadata**

- Why Is Metadata Important
- Types of Metadata
  - Index Entries
  - Conditional Processing Attributes
  - Importance, Status, and Translate Metadata Attributes
  - Topic Metadata
  - DITA Map Metadata
- Custom Metadata
- Metadata Inheritance
- To Wrap Up
- Metadata Checklist

## **Chapter 9 Conditional Processing**

- Conditional Processing Attributes
- Creating a Conditional Processing Scheme
  - Example of a Conditional Processing Scheme
- Applying Conditional Processing Attributes
  - Excluding and Including Content
  - Flagging Content
  - Improving Retrievability by Applying Metadata to Topics and DITA Maps
- Multiple and Compound Conditions
  - Multiple Conditions
  - Compound Conditions
  - Processing Logic for Multiple and Compound Conditions
- Identifying Applied Conditional Values
- Testing Your Scheme
- To Wrap Up
- Conditional Processing Checklist

## **Chapter 10 Content Reuse**

- Benefits of Reuse
- Ways to Reuse Content
- Reusing Elements by Using Content References
- Reusing Topics
- Reusing DITA Maps

Reusing Content from Non-DITA Sources

Writing for Reuse

Deciding Which Content to Reuse

Step 1: Analyze Your Content

Step 2: Identify Duplicate and Near Duplicate Content

Step 3: Address the Duplication

Step 4: Reorganize and Rewrite for Reuse

Step 5: Implement the Reuse Strategy

Track Your Reuse

To Wrap Up

Reuse Checklist

## **Part III: Converting and Editing**

### **Chapter 11 Converting Content to DITA**

Conversion Goals

Create a Pilot Team

Conversion Process

Step 1. Assess the State of Your Content

Content Analysis Worksheet

Step 2. Plan and the Conversion

Scheduling the Conversion

Converting the Content In-House or Hiring a Vendor

Staffing Your Conversion Team

Deciding on a Conversion Strategy

Defining your XML Standard

Establishing Graphics Formats

Establishing DITA File Requirements

Deciding What DITA Topic Types You Need

Establishing an Architecture for Your DITA Maps

Handling Special Structures in Your Source Files

Step 3. Prepare the Content for Conversion

Conversion Workshops

Step 4. Convert Your Source Files

Step 5. Address Postconversion Issues

Phase 1: Address <required-cleanup> Elements

Phase 2: Fix Maps and Linking

Phase 3: Improve Topics

Phase 4: Check for Markup Problems and Do Code Reviews

Phase 5: Exploit DITA

Step 6. Evaluate the Conversion Process

To Wrap Up

Conversion Sizing Table

## **Chapter 12 DITA Code Editing**

- Code Reviews

  - Code Review Benefits

- Identifying Code Reviewers

- Limiting the Scope of the Review

- Preparing for Code Reviews

  - Using Special Style Sheets for Revealing Problems in the Markup

- Performing a Code Review

  - Step 1: Schedule the Code Review

  - Step 2: Submit the DITA Topics for Review

  - Step 3: Review the DITA Markup

  - Step 4: Discuss Review Findings

  - Step 5: Complete the Code Review

- Code Reviews for Content Not in Topic Form

- To Wrap Up

- Code Review Checklist

## **Chapter 13 Content Editing**

- Defining, Scheduling, and Submitting Content Edits

  - Defining the Types of Content Edits

  - Scheduling the Edits

  - Submitting Content for Editing

- Providing Editorial Feedback

  - Inserting Draft Comments

  - Inserting XML Comments

  - Tracking Changes

  - Comparing Original and Edited Files

- Editing the Content in DITA Topics and Maps

  - Editing DITA Topics

- Editing the Output

- To Wrap Up

- Content Editing Checklist

---

# Acknowledgments

A single page could never express our gratitude to the family, friends, and colleagues who contributed to this effort. But we're not quitters, so we'll attempt to thank everyone who made this book possible.

We appreciate the camaraderie and support from the DITA community. The user groups, forums, and conferences have helped to shape our DITA knowledge. We want our peers and colleagues to understand how valuable they have been.

We are fortunate to have such leaders in the DITA community as reviewers: Thank you to Don Day, chair of the OASIS DITA Technical Committee, for taking the time to be one of our reviewers. Your DITA knowledge is vast and we appreciate the help. Thank you to Yas Etesam for lending us your detailed technical knowledge of DITA and your experience leading DITA implementations at such diverse companies. Amber Swope brought a more complete view for converting content to DITA. Heather Crognale is the best kind of friend, the one who reviews your manuscript and still sends you a Christmas card. Because of her years of experience as a DITA editor and her eagle eye as a copy editor, she has helped others to become DITA editors. Evelyn Eldridge is already a fan of editing in DITA, and her contribution to this book will spread the enthusiasm.

Thank you to Rob Lee for designing the icons. Without his contribution, we would never have known that teal is the new *in* color.

We would also like to thank Janine Trakhtenberg and Just Systems whose XMetaL authoring tool we used to produce the sample DITA files and book examples.

Of course, no book would be worth the reading if that book weren't given a good working over by editors. Good editors simply make writers look better. So thank you to Shannon Rouiller, who provided expert advice on short descriptions, topic-based writing, and other

chapters. Shannon's thoughtful and kind advice helped us turn rough drafts into coherent information. Thanks to Elizabeth Wilde who showed us a new direction for the topic-based writing chapters and helped us dig deeper to find better examples, better tone, and better organization. And thanks to Marianne White, who helped us improve the consistency of our terminology, improve our examples, catch all those embarrassing nits, and improve the flow of the chapters.

Finally, a special thanks to our families and friends. It'll take a lot more than a few words to make up for all the late nights, the missed weekends together, and the constant dinner conversation about the default behavior of related links. Throughout this effort, our families and friends have become unwitting DITA experts. Despite the fact that knowing DITA is a very marketable skill, we realize that they put up with quite a lot. Their support, encouragement, and steady supply of caffeine made all the difference.

---

## About the Authors



Laura Bellamy is an Information Architect at VMware, Inc. and a technical communications instructor at University of California Santa Cruz Extension. Laura has been a long-time DITA champion, working at IBM during the adoption and proliferation of DITA. Throughout her career she has worked on many facets of DITA implementation and now dreams in XML.



Michelle Carey is a technical editor at IBM and a technical communications instructor at University of California Santa Cruz Extension. Michelle has taught IBM teams and users' groups about best practices for authoring in DITA, topic-based writing, writing for translation, editing user interfaces, and writing effective error messages. She is also a coauthor of the book *Developing Quality Technical Information*. Michelle loves to ride motorcycles and mountain bikes, herd cats, and diagram sentences.



Jenifer Schlotfeldt is a project leader, information developer, and technical leader at IBM and a technical communications instructor at the University of California Santa Cruz Extension. She has been authoring, testing, and teaching DITA at IBM since 2003. She has converted documentation to DITA, authored new content in DITA, contributed to new DITA specializations, and created many training materials for different facets of DITA authoring.

---

# Introduction

So, finally, you've decided to write your content in DITA.

First, congratulations! For most companies or organizations, deciding whether to adopt Darwin Information Typing Architecture (DITA) as an XML authoring methodology is an arduous journey. We're strong supporters of the DITA standard and are confident that your investment to move to DITA can improve the technical information that you create for your products, services, or technologies.

Second, now what are you going to do? How do you start implementing DITA? What do you need to know before you start writing? What best practices have many of us in the technical writing community established? What are the gotchas and pitfalls of authoring in DITA? What's the best way to learn about DITA?

Don't panic: We've helped to educate many teams in our own companies and communities and promise you that learning how to semantically tag content in XML is not the hard part. Learning how to best use (or not use) the DITA elements, DITA maps, and topic types to fit your content and your organization is the real challenge.

The default DITA standard includes more than 400 elements, and those elements contain attributes. You'll need to consider these questions before you get started:

- Should you use all of the DITA elements and attributes?
- Which features do you need to understand to get going?
- Which features can you wait to implement after you've spent more time working with DITA?
- What are the guidelines that you should follow?

This book is for users who have made the decision to use DITA and are looking for advice from experienced DITA authors, editors, and information architects about how to write effective technical information in DITA.

We decided to write this book to fill an information gap in the available DITA information. Some DITA books and educational courses tell you what each DITA element represents and what it is used for. However, they don't always tell you how to best create effective content for these elements or how to organize that content.

For example, current DITA education defines what a <shortdesc> element is, but that education doesn't show you how to write effective text to go in that element.

We've spent years evaluating, testing, and writing best practices for technical writers at our companies. Our recommendations go beyond defining elements: We'll show you how and when to use an element, how to write effective text for that element, and even when not to use specific elements and attributes.

One consistent question that we hear at conferences and community discussions is, "How did you agree on best practices and write those guidelines?" The answer is years and years of trial and error until we refined our guidelines to a set of standards that help us to create industry-leading technical information.

For example, we've presented our short description best practices at DITA conferences and user group forums and meetings. At one presentation, a colleague said, "I wish someone would sell us these best practices so we don't have to spend months writing our own guidelines." Good idea!

Writers, editors, information architects, and even managers will find helpful guidelines and best practices for writing, organizing, and editing DITA content, and converting non-DITA content to DITA topics:

- Technical writers will learn how topic-based writing in DITA can help them create more effective information.
- Editors will learn about new ways to ensure the quality of the information.
- Information and XML architects will get practical advice about which DITA elements and features to implement.
- Project managers will find roadmaps and checklists to help them coordinate the conversion to DITA.
- Technical writing managers will find information about roles and resources required for converting content to DITA.

We created these examples and best practices by using out-of-the-box DITA authoring and processing tools. We used DITA 1.1 in the XMetaL authoring tool and produced PDF and HTML output by using the DITA Open Toolkit. In some cases, we highlighted features that are unique to DITA 1.2. You, too, can create effective topic-based information by using the default settings in DITA.

You can find a number of resources to help you learn DITA basics, and the DITA community is a wonderfully supportive group.

In some chapters, we refer to two other books that cover similar topics: *The IBM Style Guide: Conventions for Writers and Editors* by DeRespinis et al. (ISBN: 9780132101301), and *Developing Quality Technical Information: A Handbook for Writers and Editors* by Hargis et al. (ISBN: 9780137034574), both published by IBM Press (Pearson Education, Inc.).

We hope that you find these guidelines helpful. Now, let's have some fun with DITA!

## **EBOOK BUNDLE VERSION**

Direct from industry experts in information development comes an eBook bundle that combines three titles for technical writers, editors, and information architects: ***Best Practices for Technical Writers and Editors: DITA, Quality, and Style (Collection)*** by **IBM Press**. This set of titles is the most comprehensive collection of resources available for technical communicators.

***DITA Best Practices*** covers Darwin Information Typing Architecture (DITA)—today's most powerful toolbox for constructing topic-based information. ***The IBM Style Guide*** provides complete, proven guidelines for writing consistent, clear, concise, and easy-to-translate content. ***Developing Quality Technical Information*** is the definitive guide to developing outstanding technical documentation—for the web and for print.

## Topic-Based Writing in DITA

What's a topic other than a conversational piece? In technical information, a *topic*, which is sometimes called an *article*, has a title and some content. A topic has just enough content to make sense by itself but not so much content that it covers more than one procedure, one concept, or one type of reference information.

Although a topic should be self-contained, it shouldn't live alone. For information delivered with most technical products, services, or technologies, a topic needs a home in a larger, organized collection of topics. That collection can then be packaged and delivered in an output format such as HTML web pages, online help, or a PDF manual.

Darwin Information Typing Architecture (DITA) is specifically designed to support topic-based writing. With its modular-based architecture, semantic XML elements, and powerful linking features, DITA can help you create and maintain topic-based, or componentized, technical information.

Writing topic-based information has advantages for both the users of your products and for your writing teams. However, separating your information into different topic types isn't enough. To ensure that your information meets the needs of your users, you need to understand how to write minimalist, task-oriented information.

### Books, Topics, and Webs of Information

Books are great for some things, such as mystery novels, complex scientific concepts, and philosophical ponderings. However, books aren't the best vehicle for delivering targeted technical content to users who have real-world tasks to complete.

Information organized in a narrative book model typically has these characteristics:

- Beginnings, middles, and endings intended to be read linearly from beginning to end
- Chapters or sections that often mix task, conceptual, and reference information

Although the narrative book model works well for novels and some nonfiction books, it's not well suited for technical information that's delivered with a product. When it's time to adjust the valves on the motorcycle, users don't want to read a novel. They want to open the motorcycle manual, find that one specific task, and move on.

A topic is a self-contained unit of information. An effective topic covers only one subject. Each topic is long enough to make sense on its own, but short enough to stick to one point without expanding into other subjects.

A topic typically answers one of these questions:

- How do I do it?
- What is it?
- What is the process?
- How do I fix it?

Because most users need only a small amount of information at a time, you should create information that can answer specific questions discretely without requiring extensive reading across large amounts of interconnected content.

Separating information into discrete topics helps you to:

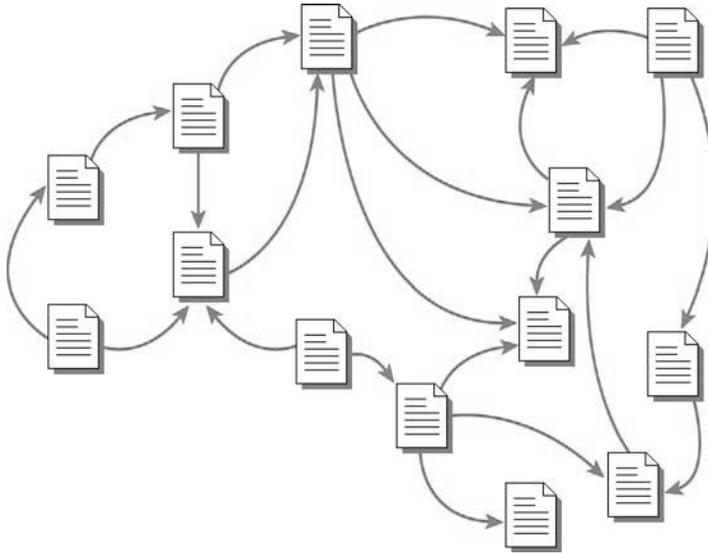
- Design new information more easily and consistently
- Eliminate unimportant or redundant information
- Identify common or reusable topics or pieces of content in topics

All topics, regardless of their purpose, have the following characteristics:

- Meaningful titles
- Ability to stand alone from other content
- Logical organization
- Links to other topics that contain related information

You organize and link topics to create a coherent web of information, as shown in Figure 1.1. Eventually, the output of this collection of topics might be an online help system, a PDF manual, or a website. Although a web of topics might seem chaotic at first, DITA can help you organize and link topics so that users don't get lost.

By writing your information in discrete topics, organizing those topics into logical collections, and then linking related topics to each other, you can create a web of information that is easy to navigate, easy to understand, and easy to consume.



**Figure 1.1** Linked and organized topics that form a web of information.

## **Advantages of Writing in Topics for Writing Teams**

Creating technical information in topics not only helps your users find and use your information more easily, but topic-based information also has significant benefits for writers, information architects, and editors.

### **Writers Can Work More Productively**

In your organization, several writers might work on content for related features or functional areas. To be more productive, each writer can contribute specific topics that support those related features or functions.

For example, for a complex installation guide for an enterprise database system, each writer can own some of the installation information. One writer might own information about setting up security, and another writer might own information about planning for the installation.

When you write in topics, it's easier and more productive for multiple writers to contribute to the larger set of information.

### **Writers Can Share Content with Other Writers**

The larger a file is, the more difficult it is for you and other writers to work on content in the same file. Imagine a file that is 50 pages. Every time you work on that file, 50 pages of content are locked and unavailable for other writers to work on.

With topics, you can work on smaller units of content at any time because each topic is one file, which allows more writers to have access to larger portions of information.

### **Writers Can Reuse Topics**

Organizations save time and resources by reusing content. You can reuse topics for multiple products, for different audiences, and for multiple information sets and output formats.

For example, you might include the same topic in both a book and a help system or share topics among product libraries. If the shared topic has content that isn't appropriate or required for every product, you can use conditional processing attributes to remove content that's not applicable rather than write and maintain two topics with nearly identical content.

### **Writers Can More Quickly Organize or Reorganize Content**

Information designed with a narrative flow or book structure doesn't enable you to quickly rearrange information. For example, if you create separate task topics for assembling a motorcycle engine, you can easily change the order of those task topics if the motorcycle engine's design changes.

### **Reviewers Can Review Small Groups of Topics Instead of Long Books**

Instead of asking editors, information architects, or technical experts to review long books, you can submit small sets of topics or even single topics to reviewers throughout the development cycle of the product. You're more likely to get better feedback if the reviewer can read through a handful of shorter topics than have to comb through a 100-page chapter.

### **DITA Topic Types**

Users of technical information often need at least three types of content: procedures, background or conceptual information, and quick reference information. What most users don't need is a jumbled mix of those three types of information—for example, procedures buried in a long section of conceptual information or a table of miscellaneous commands thrown in the middle of a procedure.

Figure 1.2 shows how mixing information types can make specific pieces of information difficult to find. The topic uses a title that seems to indicate that the information is conceptual, and the topic does describe how nuclear energy is created. However, the topic then digresses into a task that describes how to connect an espresso machine to a nuclear reactor. And to make matters worse, the topic also contains a table of commands, none of which are related to how energy is created from nuclear fusion. Most users do not expect to find task information or even a list of commands in a topic called "Nuclear fusion as a power source."

## Nuclear fusion as a power source

Fusion power is the process by which multiple atomic particles join together to form a heavier nucleus. This is the power source for the Exprezzoh 9000N.

This process creates the release or absorption of energy. Iron and nickel nuclei have the largest binding energies per nucleon of all nuclei and therefore are the most stable. The fusion of two nuclei lighter than iron or nickel generally releases energy while the fusion of nuclei heavier than iron or nickel absorbs energy.

The opposite process is called nuclear fission. However, because the Exprezzoh 9000N requires so much energy, it uses the nuclear fusion process.

You can connect the Exprezzoh 9000N to any operating nuclear fusion plant in a 200 mile radius of your home. To connect to a fusion reactor:

1. Obtain the required permits from your local nuclear federal agency.
2. Set up the monitoring system that is required by the federal agency.
3. Ensure that the nuclear power source light is green.

The power source display panel is near your circuit-breaker panel for your house. The nuclear power display panel was installed when you installed the nuclear reactor piping cables.

4. Plug in the Exprezzoh 9000N to the fusion reactor power cord.
5. Start the Exprezzoh 9000N by turning the ON/OFF switch to the ON position.

After you connect the nuclear reactor, you can monitor power flow from your personal computer. You can also redirect some of that power to other appliances.

| Task  | Command  | Example  |
|---|--|--|
| Get reports at various intervals to view power consumption. | <pre>exprezzoh report powerLevel time</pre> <p>For <i>time</i> ,you can specify to get reports every hour, every 12 hours, every day, every week, or every month:</p> <p><b>hour</b><br/>A report is created every hour.</p> <p><b>12 hour</b><br/>A report is created every 12 hours.</p> | <pre>exprezzoh report<br/>powerLevel 12 hour</pre> |

**Figure 1.2** A topic with conceptual, task, and reference information.

To make it easier to create and deliver information that effectively separates content by type and purpose, DITA provides three main topic types: concept, task, and reference:

- A task topic describes one procedure.
- A concept topic defines what something is or how a process works.
- A reference topic contains one type of reference information that users might need as they perform tasks.

Table 1.1 shows the differences between concept, task, and reference topic types by their titles.

By separating content by type, you prevent users from wading through information that they don't need. For example, when you want to install a home entertainment system for the first time, you don't need to know about all the buttons on the remote control (reference information). Rather, you need the installation instructions. By separating the reference information from the task information, users can more quickly install their system.

Separating content by type also helps specific users find the information they need. For example, novice users are more likely to need conceptual information, whereas experts probably go straight to the procedures and reference material and might not bother with the concepts.

**Table 1.1** Examples of Topic Types by Title

| Concept Topic Title        | Task Topic Title                           | Reference Topic Title                          |
|----------------------------|--|--|
| User roles                 | Create user roles                          | Supported types of roles                       |
| High-definition television | Install a high-definition television set   | Television set accessories                     |
| Espresso                   | Make espresso drinks                       | Espresso drink ingredients                     |
| Cat behavior               | Herd cats                                  | Domestic cat varieties                         |
| Databases                  | Configure databases for enterprise systems | Database types                                 |
| Photography                | Take photographs of landscapes             | Digital camera models and compatibility matrix |

## Task Orientation

Separating your content by type isn't always sufficient to make your information address the needs of your users. To help users accomplish real-world goals by using your products, you must create task-oriented topics. For example, the users of your products might want to accomplish the following real-world goals:

- Processing loan applications for a bank
- Reducing power costs by installing solar panels on rooftops
- Increasing revenues by making business processes more efficient
- Making electronic medical records available to physicians
- Manufacturing audio components for cars
- Setting up enterprise email systems so that employees can be more productive

The job of the technical writer, editor, and information architect is to create information for products that help users accomplish these goals. The goal is not simply to describe how the product works.

To create effective task-oriented information, follow these guidelines:

- Focus on the goals of the user, not the way the product works.
- Write from the user's point of view and write in active voice.
- Target the appropriate audience.
- Tell users why they need or should perform the task.
- Break up large or complex tasks into shorter subtasks and organize task topics in logical order.
- Don't bury a task in conceptual or reference information.

By applying the principles of task-oriented writing to your information, you can write your topics according to what tasks users perform rather than by the way the product works or by the way it's designed.

For more information about task-oriented writing, see "Task Orientation" in *Developing Quality Technical Information* by Hargis et al.

## Task Analysis

When human factors engineers design new products, they do a task analysis so that they can understand the goals of their users and how those users want to use the product. Technical writers can also do a task analysis to understand how users work with the product. A task analysis can help you create effective task-oriented information.

During a task analysis, you need to find as much information as you can about how users currently work or intend to work with your product. A thorough task analysis can provide the following information:

- What task topics to write
- How much supporting reference and conceptual information to provide

Do a task analysis at the beginning of the project. For example, you might do a task analysis when you create information for a new product, when you reorganize a set of information, or when you model the information for a new feature, service, or technology.



**Tip:** Project managers can also use your task analysis to scope the documentation project. Your analysis can help managers understand the size and resources required for the project.

Table 1.2 shows a task analysis of how to make an espresso drink:

**Table 1.2** Task Analysis of How to Make Espresso Drinks

| Question   | Details  | Comments   |
|--|--|--|
| What is the goal?  | To make an espresso drink.   | State the ultimate goal of the user; don't describe how the product works. |
| What tasks does the user need to perform to accomplish the goal? | <ul style="list-style-type: none"><li>• Prepare the beans.</li><li>• Load the filter.</li><li>• Configure the espresso machine.</li><li>• Add water.</li><li>• Prepare the milk.</li><li>• Turn on the espresso machine.</li></ul> | Don't worry about order yet. Just brainstorm all possible tasks.           |
| What are the mental and physical steps involved in each task?    | Mental: Decide what kind of coffee to make.<br>Physical: Grind beans, steam milk, and load filter.   | Most tasks require mental and physical steps.                              |
| Who performs the task?   | Audience: Coffee drinkers who like strong coffee drinks<br>Experience: Advanced  | Describe users in as much detail as possible.                              |
| When and under what conditions is the task performed?            | Requirements: Espresso machine must be configured and running; espresso drink ingredients must be available.<br>Limitations: User must know how to create espresso drinks.   | Describe prerequisites, limitations, or restrictions to do the task.       |

|  |  |  |
|--|--|--|
|  | Environment: Users usually make coffee in the morning and are probably sleepy.   |  |
| What are potential distractions to accomplishing the goal? | Troubleshooting: Broken power source or bean grinding problem.<br>Alternative path: Deciding to make brewed coffee rather than an espresso drink.<br>Exception path: Missing ingredients.  | Consider errors or problems that users might encounter.<br>An exception path describes the situation when something goes wrong in the process that prevents users from completing of the task. |
| What does the user need to know about the task?            | Duration: 3 minutes if the coffee beans are ground; 8 minutes if the coffee beans are not ground.<br>Complexity: Easy for advanced users; medium to difficult for users making espresso drinks for the first or second time.<br>Frequency: Daily.  | This information might impact the type of content that you include in the procedure or how you structure the task topic.   |
| What is the sequence of tasks or steps?                    | Prerequisites: Install and configure espresso machine.<br>1. Turn on the espresso machine.<br>2. Prepare the beans:<br>a. Select the coffee beans.<br>b. Grind the beans to fine or extra fine.<br>3. Load the ingredients:<br>a. Load the ground beans.<br>b. Add the water.<br>4. Allow the ground coffee to brew.<br>5. Pour the espresso into a cup.<br>6. Optional: Add sugar or other ingredients. | Organize the tasks or steps in the proper order.   |
| What is the expected result?                               | Make a perfectly crafted espresso drink.   | State the results that users will expect to see or accomplish.   |

The result of the analysis is that you have the following information:

- An understanding of what tasks users must complete to achieve a goal.
- An outline of what steps it takes to perform a task. Use this outline to create your DITA topics.

Although a task analysis might seem time consuming, the effort can likely save you time over the entire release cycle of your product. Use the task analysis form that appears at the end of this chapter to do an analysis for your product, service, or technology.

You can use professional modeling tools to do a task analysis, such as UML applications or the DITA-aware IBM Information Architecture Workbench. You can also use a spreadsheet or pen and paper to track the analysis questions and responses.

## Minimalist Writing

No discussion of topic-based writing would be complete without a few words...very few...about minimalist writing. By following minimalist writing principles, you can create more effective topics. In minimalist writing, you should provide only the information that users need, when they need it, and nothing more.

You can find many excellent books and articles that describe minimalist writing in more detail, but remember these important principles: know your audience, remove nonessential content, and focus on user goals.

### Know Your Audience

You must understand your users' level of expertise with the product, service, or technology. Do a task analysis so that you know exactly what information users need to accomplish real goals.

Analyze customer support feedback and conduct usability tests to learn about your users:

- What are users likely to know about your product or technology? For example, if your product describes how to create web pages, do you need to explain the basics of browsers?
- What goals do they want to accomplish?
- Will they understand the terminology used for your product? For example, will most of your users who are experienced with search engines know what you mean by "Boolean operators"?
- How much help will they need to resolve problems with the product? How much troubleshooting information should you provide?

### Remove Nonessential Content

As a technical writer, you become an expert in the product, service, or technology that you document. Although you might understand your product well, consider what information is essential for your users. Don't create information that users don't need or care about.

This advice might sound like a no-brainer, but you often read technical information that describes the toolbar icons of a simple software product or a step in a task that says, "Type your name in the name field." If your product requires explanation of the toolbar or how to enter a name in a name field, consider improving the product interface rather than creating topics that will probably never be read.

### Focus on User Goals, Not Product Functions

Avoid writing task topics that are solely about how the product works. For example, instead of writing a task topic called "Using the User Profile dialog box," which focuses on how the product works, create a task called "Changing user profiles," which focuses on a real goal.

Even for complex enterprise products, such as database or search engine systems, avoid writing lengthy discussions about how the product works. Your users want only enough information to help them set up and use their systems or products.

For example, instead of writing long chapters about how authentication works, start with tasks that show users how to set up security, and explain the options and the benefits of setting up security for various scenarios as they progress through the tasks. You can provide some concept topics, but provide just enough to get users started. Very few readers will be patient enough to read 25 or 50 pages of conceptual material before they start a task.

For more information about writing consistent, minimalist content, see *The IBM Style Guide: Conventions for Writers and Editors* by DeRespinis, et al.

## To Wrap Up

DITA provides a flexible, yet rigid, framework that helps you create effective technical information. DITA is flexible in that it helps you reuse, reorganize, and create content quickly. It's rigid in that you must be disciplined enough to adhere to the principles of good topic-based writing to take advantage of the benefits of DITA.

Writing task-oriented DITA topics takes some practice, but when you do create those perfect topics, your users can:

- Find the information they need faster
- Accomplish their goals more efficiently
- Read only the information they need to read

Writing in DITA not only benefits your users, but it also benefits you as a writer. By using DITA to create topic-based information, you can:

- Maintain and reuse topics more effectively
- Organize or reorganize topics more quickly
- Share and distribute the work on topic files more easily, which increases writer productivity

Topic-based information also starts with a thorough knowledge of your users and their goals. Do a comprehensive task analysis before you write anything. Don't be afraid to spend more time on planning your information and analyzing the tasks that users will want to do with your product.

By writing your content in discrete, task-oriented DITA topics, you can create useful, coherent, and retrievable technical information.

## Topic-Based Writing Checklist

| Guideline   | Description  |
|---|--|
| Check that each topic contains only one type of information.              | Use the different topic types in DITA to separate task, conceptual, and reference information.   |
| Check that each topic is self-contained.                                  | Eliminate text that points to other topics (other than the occasional cross-reference), text that points to sections in the topic, such as “The next section describes table spaces,” or topics that serve merely to glue other topics together.                                 |
| Check that topics don’t cover too much information.                       | For example, writing one topic to describe even a single commercial aircraft model is too much information for one topic. Instead, break up the topic into more concept topics and subtopics, such as “Engines,” “Passenger seating,” “Pilot console,” and “Electrical systems.” |
| Ensure that your content is task oriented.                                | Task orientation means that your content focuses on real user goals, not product functions or features. Don’t focus on pseudo-tasks such as “Using the email feature.” Using something isn’t a real-world goal. “Sending a document through email” is a real goal.               |
| Do a task analysis to decide what information your topics should contain. | Technical information is effective only if it answers users’ questions quickly and completely. Ensure that you understand the level of expertise of your users and their goals.  |
| Follow minimalist guidelines.   | <ul style="list-style-type: none"><li>• Know your audience.</li><li>• Eliminate nonessential content. Write only the content that users need. Don’t paper the product.</li><li>• Focus on user goals, not product functions.</li></ul>   |

# Task Analysis Form

Use this form to do a task analysis on a new product, feature, or component.

| Question   | Details           | Comments |
|--|-------------------|----------|
| What is the business goal?                                       |                   |          |
| What tasks does the user need to perform to accomplish the goal? |                   |          |
| What are the mental and physical steps required for each task?   | Mental:           |          |
|  |                   |          |
|  | Physical:         |          |
| Who performs the task?   | Audience:         |          |
|  |                   |          |
|  | Experience:       |          |
|  |                   |          |
|  | Role:             |          |
|  |                   |          |
|  | Authority:        |          |
| When and under what conditions is the task performed?            | Requirements:     |          |
|  |                   |          |
|  | Limitations:      |          |
|  |                   |          |
|  | Environment:      |          |
| What are potential distractions to accomplishing the goal?       | Troubleshooting:  |          |
|  |                   |          |
|  | Alternative path: |          |
|  |                   |          |
|  | Exception path:   |          |
| What does the user need to know about the task?                  | Duration:         |          |
|  |                   |          |
|  | Complexity:       |          |
|  |                   |          |
|  | Frequency:        |          |
| What is the sequence of tasks?                                   |                   |          |
| What is the expected result?                                     |                   |          |