

LYNN DENTON
JODY KELLY

DESIGNING,
WRITING &
PRODUCING
COMPUTER
DOCUMENTATION

JAY RANADE, CONSULTING EDITOR



Designing, Writing, and Producing Computer Documentation

**Lynn Denton
Jody Kelly**

McGraw-Hill, Inc.

**New York St. Louis San Francisco Blue Ridge Summit, Pa.
Auckland Bogotá Caracas Lisbon London Madrid
Mexico Milan Montreal New Delhi Paris San Juan
São Paulo Singapore Sydney Tokyo Toronto**

Library of Congress Cataloging-in-Publication Data

Denton, Lynn.

Designing, writing, and producing computer documentation / by Lynn
Denton and Jody Kelly.

p. cm.

Includes index.

ISBN 0-07-016412-6 (H) ISBN 0-07-016417-7 (P)

1. Electronic data processing documentation. I. Kelly, Jody.

II. Title.

QA76.9.D6D46 1992

808'.066005—dc20

92-20327

CIP

Copyright © 1993 by McGraw-Hill. Printed in the United
States of America. Except as permitted under the United States
Copyright Act of 1976, no part of this publication may be reproduced
or distributed in any form or by any means, or stored in a data base
or retrieval system, without the prior written permission of the
publishers.

1 2 3 4 5 6 7 8 9 0 HALHAL 8 9 4 3 2 1 0 9

ISBN 0-07-016412-6 (H) ISBN 0-07-016417-7 (P)

Notices

Apple, the Apple logo, Macintosh, Imagewriter, LaserWriter, MultiFinder, and Stackware are trademarks of Apple Computer, Inc., registered in the United States and other jurisdictions. HyperCard, HyperTalk, and MacPaint are trademarks of Claris Corporation registered in the United States and other jurisdictions.

IBM is a registered trademark of International Business Machines Corporation.

Information contained in this work has been obtained by McGraw-Hill, Inc. from sources believed to be reliable. However, neither McGraw-Hill nor its authors guarantees the accuracy or completeness of any information published herein and neither McGraw-Hill nor its authors shall be responsible for any errors, omissions, or damages arising out of use of this information. This work is published with the understanding that McGraw-Hill and its authors are supplying information but are not attempting to render engineering or other professional services. If such services are required, the assistance of an appropriate professional should be sought.

*For more information about other McGraw-Hill materials,
call 1-800-2-MCGRAW in the United States. In other
countries, call your nearest McGraw-Hill office.*

Designing, Writing, and Producing Computer Documentation

Acknowledgments

The authors wish to thank International Business Machines Corporation for permission to use the IBM host publishing system, BookMaster 3.0, to produce the final manuscript of this book. To the reviewer of our early drafts, Peggy Newfield, we owe great thanks for making the book much better than it was.

We also wish to thank David McMurrey, Karen K. Travis, Lance C. Amundsen, and Tom T. Beard for special help. Finally, we owe more than can be repaid to our families and friends, especially to Holly Atlas.

Introduction

This book is designed to help writers in the computer industry make their product documentation more useful, attractive, and accessible to their paying customers by building in quality from the beginning.

To produce good documentation, a technical writer has to play many roles: market researcher, library designer, contract negotiator, interviewer, technical researcher, scheduler, financial planner, customer advocate, task analyst, peer reviewer, writer, editor, project manager, moderator of review meetings, planner for translation and foreign distribution, book designer, layout artist, typesetter, legal researcher, usability test coordinator, indexer, glossary writer, distributor, and proofreader.

Writers in one- or two-person writing shops might do all of these jobs and more. Those in larger shops might specialize at first and then rotate into other jobs over time. Mastering all the skills needed to produce quality computer documentation is not easy; this book presents a comprehensive “crash course” in the field.

If you are a new or prospective technical writer, or an experienced writer interested in improving the quality of your documentation, this book is for you. Computer designers, engineers, programmers, and product planners—anyone who writes technical information that others must read and use—will also find many helpful tips and techniques.

Although this book is not designed specifically as a textbook, it includes enough practical examples to serve as a reference for students and instructors of technical writing.

Contents

Acknowledgments	ix
Introduction	xi
Chapter 1. Achieving Quality through Library Design	1
Support for Customers' Tasks	1
Defining Quality in Computer Documentation	7
Other Influences on Quality	10
Chapter 2. Analyzing Your Audience and Purpose	17
What Is an Audience Analysis?	18
Assessing Technical Background	21
How Do You Obtain Information?	25
Summary of Audience Analysis	26
Chapter 3. Getting Organized	29
Finding the Source Information	29
Understanding the Product	33
Organizing the Source Information	35
Chapter 4. Dividing the Information into Books	39
What Do You Do with Your Audience Analysis?	39
Organizing Your Book	40
Outlining a Book	46
Using Rhetorical Patterns	49
Chapter 5. Writing Information Components	57
Writing Components and Nodes	58
Putting the Nodes Together	66
Revising Your Writing	68
Editing Your Writing	80

Chapter 6. Presenting the Information	85
Front Matter	86
Body	91
Back Matter	94
 Chapter 7. Planning Layout	 101
Who Does the Design work?	101
Principles of Design	104
Planning Page Layout	105
Planning for Retrievability	112
Selecting Typefaces and Sizes	122
Graphics	126
Conclusions	138
 Chapter 8. Packaging and Publishing the Book	 139
Packaging Your Book	140
Choosing Slipcases, Dust Jackets, and Covers	144
Selecting a Printing Process	147
Reviewing and Proofing	151
Conclusions	152
 Chapter 9. Producing Online Documentation	 153
Differences Between Online and Printed Documentation	155
Design Principles of Online Documentation	158
Types of Online Documentation	160
Tutorials	161
Softcopy Books	163
Help Panels	167
Hypertext	172
 Chapter 10. Reviewing and Testing for Quality	 179
Guidelines for Reviewing and Testing	179
Deciding Whether to Review or Test	180
Preparing for Reviewing	181
Preparing to Test Printed and Online Information	185
Testing Printed Information	190
Testing Online Information	194
Conclusions	199
 Appendix A. Style Guides	 201
Punctuation	202
Spelling	202
Highlighting	203
Numbers	203
Lists and Tables	203
Word Usage and Terminology	203

Appendix B. Sample Chapters from a Design Guide	205
Front-Matter Components	205
Back-Matter Components	213
Other Hardcopy Elements	220
Copyright and Trademark Guidelines	228
Writing Style, Usage, and Terminology	233
Index	253

Achieving Quality through Library Design

Writers in the computer industry don't set out to produce books that overwhelm, bore, confuse, and irritate customers. When these results occur, writers tend to blame insufficient time, training, or equipment; inadequate audience and market analyses; or red tape of various kinds, including poor communications among people and politically motivated decisions. Some companies, however, produce good books despite these conditions, because their writers understand quality and use a well-defined process to achieve it.

This chapter offers an introduction to task analysis, a discussion of library design, a definition of quality, and a discussion of various influences on the quality of computer documentation.

Support for Customers' Tasks

High-quality computer documentation is designed, first and foremost, to support the *tasks* that buyers of the product want to perform. Customers might or might not take an interest in the details of the various functions, but they always want clear information on how to get their work done when they use the product.

For example, a common task for customers using a word processing or a desktop publishing program is to place the page number on the page: they want to know how to position it at the top or bottom of the page and at the right, center, or left of the line. Good documentation explains in one place

all the steps necessary for customers to place the page number exactly where they want it. Less effective documentation might bury the page number under a discussion of running headers and footers, discuss top and bottom placement in a different section from right, center, or left placement, or neglect to mention that page numbering is a built-in function that can't be changed. Flaws like these result from failing to think about how customers are going to use the product and what tasks or jobs they want to do. In a sense, *everything* is a task.

Universal tasks

The following universal tasks associated with software products can accommodate even conceptual and descriptive information:

- *Evaluation.* Deciding whether to purchase and install the product
- *Planning.* Deciding on the type, number, and location of the products to install, as well as the relationships among the products
- *Installing.* Setting up the product for use
- *Administration.* Managing the resources of the product, including its users
- *Operation.* Starting, stopping, and maintaining the product
- *Tailoring.* Tuning the product for specific situations
- *Programming.* Designing, coding, compiling, running, debugging, and testing programs written for use with the product
- *Diagnosis.* Recovering from error conditions
- *End use.* Performing end-user tasks with the product

(from *Information Development Guideline: Task-Oriented Libraries* published in 1986 by IBM)

Using these universal tasks can help you group similar information into one book or, for a one-book library, into the same section of the book. You should also ensure that the subtasks present procedural information that helps customers achieve their objectives. (For an example of a detailed task analysis, see “Detailed task analysis for an installation node” on page 60.)

Library design

A good first step in achieving high-quality computer publications is library design. A *library* is a set of books supporting a computer product. The *design* of a library includes determining how much and what type of information to present, analyzing the audience, dividing the information into books,

deciding on the format, providing a design guide to ensure consistency throughout the library, and then developing a publications plan.

Planners, managers, or experienced writers usually design the library for a new product, but less-experienced writers should also understand the principles involved. Even if the product requires only one book, the process of designing a library is still useful because it ensures that the information is organized appropriately.

Determining the information to present

Before you can decide how much and what type of information the customer needs, you must thoroughly understand the product. Study the marketing objectives and technical specifications of the product to find answers to the following questions:

- Is it a new product or an update of an existing product?
- Is it a large-scale product for an entire system, a small single-purpose product, or something in between?
- What is the main function it performs?
- How many subfunctions does it contain and what are they for?
- How complex are the subfunctions?
- Who will use the product and what will be their most common tasks?
- How many ways can the product be used (selecting from menus, clicking on icons with a mouse, entering commands, and writing programs, for example)?
- What makes this product stand out among competitors' products?

With these questions answered, determine how many of the universal tasks listed on page 2 are relevant to the product:

- *Will customers need information on evaluating the product and deciding whether to buy it?* For a new product, describe the features that no competing product provides. For a new release of an existing product, explain the enhancements added since the last release.
- *Will customers need information on planning the number and location of each installation?* For example, most networking and communications products require extensive planning because the interactions among different types of hardware and software are quite complex.
- *Will customers need information on installing the product?* For a simple, automatic install, they might need to know only the command to type, but for a large, complicated product, they might require extensive assistance.

- *Will customers need information on administrative tasks?* Product resources that need to be monitored and maintained might include users, data, memory, programs, network nodes, and workstations.
- *Will customers need information on daily operations?* These operations might include starting and stopping the program, restarting the system after a power failure or some other emergency, maintaining records on the use of system resources, and replacing failing components.
- *Will customers need information on tailoring the product for a specific environment?* For example, a customer might want to speed up access to a resource, protect part of a system from novice users, or enhance a search facility.
- *Will customers need information on application programming?* If user-written programs can interact with the product, customers will need information on writing code in the supported languages.
- *Will customers need information on diagnosing errors and recovering from problems?* You might need to provide only a telephone number to call for service, or you might need to provide extensive problem analysis and resolution information.
- *How much information will customers need for end-user tasks?* Programmers and administrators might require little or no basic information, while general office workers might need only basic information on end-user tasks.

With these questions answered, sort the product information tasks and list all the tasks that different types of users can perform with the product. Duplicate tasks will become apparent. All users, for example, might need to log on, but you don't need to repeat the logon steps in each book in the library.

An important part of library design is deciding where to place common tasks and how to cross-reference these tasks with other books. In general, you'll use a sifting method:

- Tasks everyone must do should go in the end-user information and should be written for the novice user.
- Tasks all technical people might want to do should go in the administrative book.
- Tasks that only specialists do should go in the programming, service, or error-recovery book.
- Reference information usually doesn't contain tasks, so it can go wherever it fits best, sometimes in a book or section by itself.

The sifting method lets you group similar tasks together and keep cross-

references to a minimum. It can be a minor annoyance for customers to flip to other sections of a book, but it's a real source of irritation to send them to books they might not have.

Analyzing the audience

Analyzing the audience and dividing information into books should take place at almost the same time, since the two activities are closely related.

Each book should be designed for a specific audience, such as “experienced programmers who write COBOL programs to access a host database” or “network administrators who are responsible for setting up, configuring, and maintaining a local area network.” (Tailoring information to a specific group of product users is the subject of Chapter 2.)

Dividing the information into books

The chief library designer is usually responsible for establishing the initial division of product information into separate books. For each book, the designer usually provides an estimate of the length, an audience description, a tentative list of the topics to be covered, and a writing schedule. The designer might also point out relevant source material for the writers and provide a list of technical reviewers for each book.

During the book development cycle, the designer also monitors the books to ensure that the library doesn't contain redundant or inconsistent information. (Chapter 4 contains more information on dividing information into books.)

Deciding on format and design

Good design is *transparent*; it is invisible to the reader. The reader should be able to move comfortably through the book, basically unaware of the design elements.

If you use good design, the reader can focus on the content and on the task at hand—learning, using, understanding, or retrieving information. Your job is to get the reader into the book and out of it as quickly as possible. (Chapters 7 and 8 contain more information on the physical design and production of a book.)

Providing a design guide to ensure consistency

An important step in producing a good book or library is to construct a *design guide*—a document that defines and describes design elements, such as use of color, book components, depiction of menus and screen displays, and page size. You need this information to ensure that the design of your book or library is consistent, and the print shop you work with needs this information to produce exactly the appearance you want.

Your design guide can be very informal, just a collection of notes to yourself, or it can be formalized and circulated—especially if several people are working on your book or on related books in a product library. Close adherence to the specifications described in the design guide is the key to ensuring consistency.

A design guide is different from a *style guide*; you need both to ensure good writing, consistent style, and good design. You might already have an in-house style guide that you follow for each project. A style guide describes how to handle items such as bulleted lists, equations and calculations, abbreviations, units of measurement, paragraph indentations, presentation of artwork, copyright notices, and other stylistic matters.

Your organization might also have its own favorite grammar and writing handbook that the writers should follow. In some cases, sections on writing mechanics are part of a style guide. In addition, some style guides contain elements traditionally found in a design guide, or your organization might have a document called a design guide that includes information traditionally found in style guides. Whatever they are called, you need to be sure to have documents available to address design, style, and writing mechanics. (For an example of a style guide and a design guide, see Appendices A and B.)

Developing a publications plan

A *publications plan* is a set of specifications for the documentation that supports a product. For a small or one-book library, the plan might be brief and informal, but for a large library, the plan might be a lengthy, detailed document. Most publications plans include some or all of the following:

- A description of the product
- The packaging plan for the books—are all the books included with the product diskettes, or are some sold separately from the product? How can the books be ordered?
- Information on whether the books will be translated into other languages
- A list containing the following information about each book in the library:
 - ~ The title of the document
 - ~ The medium of the document (printed or online)
 - ~ The estimated length in pages or panels
 - ~ The writer(s) assigned to the book
 - ~ The purpose of the book
 - ~ The audience for whom the book is intended

- ~ The schedule for distributing and reviewing the outline and each draft of the book
- ~ The last date for changes to the book before it is printed
- ~ The required reviewers of the book
- ~ A description of the content expected in the book
- ~ The manufacturing specifications of the book, such as the page size and the binding method

Typically, a publications planner or experienced writer produces the plan and distributes it to all those who might be affected by it: writers, programmers, marketing personnel, manufacturing schedulers, and managers, for example.

Defining Quality in Computer Documentation

The most important indicator of quality is that the information helps customers do their work. Some other characteristics of excellent documentation are that it is easy to understand and use, it enables information to be easily retrieved, its format and visual devices are effective, it is consistent in style and tone, and it is meticulously written.

Technical accuracy and completeness

Technical accuracy means that the product works the way the documentation says it does, as based on actual experience rather than theory. To ensure that your document is technically accurate, carefully check the correctness of each of the following:

- Text and artwork representing the user interface
- The functions, commands, and routines of the product
- Parameters, input and output values, and defaults
- Error messages and recovery procedures
- Navigation paths through the product panels
- User actions and their results

Technical completeness means that the documentation covers everything the customer might want to do with the product or might need to know about it. A technically complete document might also include some or all of the following, if appropriate for the intended audience:

- Information on the fastest, safest, least expensive, or easiest way to accomplish a task

- Information on how often and when to perform certain tasks
- Reference information
- Information on preventing and recovering from unexpected results

If you can anticipate how customers might use the product in relation to other products, the documentation should give them some guidance on the probable interactions. This kind of information might be difficult for you to find, but it is very useful to customers when you can provide it. (Chapter 3 offers several ways to locate this kind of information.)

Ease of understanding and use

Your documentation should be easy for the intended audience to understand, whether the customers are novices or experts. The writing should be at a level appropriate to the audience, and you should include enough clear explanations, examples, and illustrations to enable customers to understand the information.

Documentation is easy to use only if customers can find what they're looking for and then act upon what they've found. The documentation should ensure customers' success in using the product.

A document that's easy to understand and use employs clear, concise writing that's simple and to the point. It also carefully separates introductory information from procedural information that requires the customer to take some action. Its steps are easy to understand and perform, its logical organization makes sense to the customer, and it provides careful cross-references to related information.

Retrievability

Documentation should be arranged so that customers can find specific information quickly and easily. You can improve retrievability in your documentation by including the following aids:

- A book cover or dust jacket displaying the title of the book
- Table of contents
- List of tables
- List of figures
- Chapter and section headings
- Cross-references
- Glossary
- Index
- Physical devices, such as die-cut tabs, bleeding tabs, and dividers