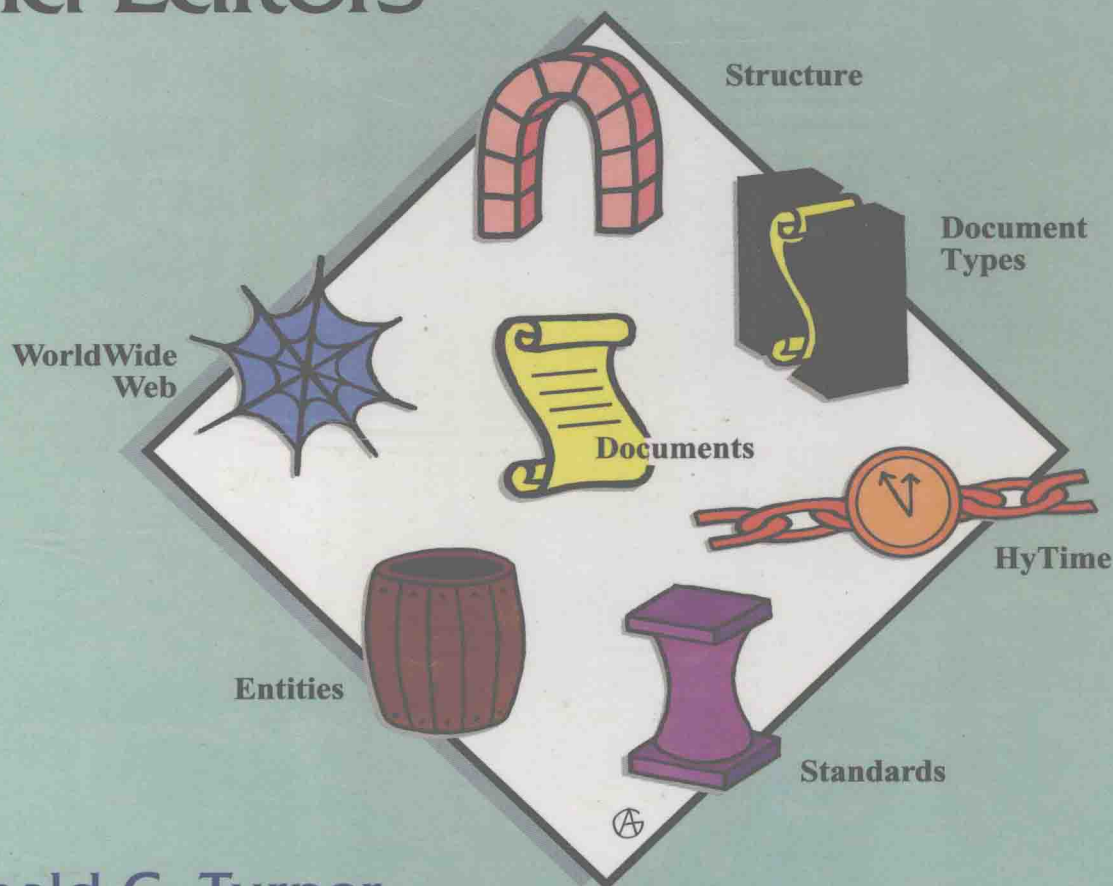


README.1ST

SGML For Writers and Editors



Ronald C. Turner
Timothy A. Douglass
Audrey J. Turner

README.1ST

SGML for Writers and Editors

Ronald C. Turner
Timothy A. Douglass
Audrey J. Turner

For book and bookstore information



<http://www.prenhall.com>



Prentice Hall PTR
Upper Saddle River, New Jersey 07458

Editorial/production supervision: *Camille Trentacoste*

Cover design: *Andrew Goldfarb*

Manufacturing manager: *Alexis R. Heydt*

Acquisitions editor: *Mark L. Taub*

Editorial Assistant: *Dori Steinhauß*



© 1996 by Prentice Hall PTR
Prentice-Hall, Inc.
A Simon & Schuster Company
Upper Saddle River, New Jersey 07458

The publisher offers discounts on this book when ordered in bulk quantities.

For more information, contact:

Corporate Sales Department, Prentice Hall PTR
One Lake Street, Upper Saddle River, NJ 07458
Phone: 800-382-3419; FAX: 201- 236-7141
E-mail: corpsales@prenhall.com

All rights reserved. No part of this book may be reproduced, in any form or by any means, without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-432717-9

Prentice-Hall International (UK) Limited, *London*

Prentice-Hall of Australia Pty. Limited, *Sydney*

Prentice-Hall Canada Inc., *Toronto*

Prentice-Hall Hispanoamericana, S.A., *Mexico*

Prentice-Hall of India Private Limited, *New Delhi*

Prentice-Hall of Japan, Inc., *Tokyo*

Simon & Schuster Asia Pte. Ltd., *Singapor*

Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*

README.1ST

SGML for Writers and Editors

The Charles F. Goldfarb Series on Open Information Management

“Open Information Management” (OIM) means managing information so that it is open to processing by any program, not just the program that created it. That extends even to application programs not conceived of at the time the information was created.

OIM is based on the principle of data independence: data should be stored in computers in non-proprietary, genuinely standardized representations. And that applies even when the data is the content of a document. Its representation should distinguish the innate information from the proprietary codes of document processing programs and the artifacts of particular presentation styles.

Business data bases—which rigorously separate the real data from the input forms and output report—achieved data independence decades ago. But documents, unlike business data, have historically been created in the context of a particular output presentation style. So for document data, independence was largely unachievable until recently.

That is doubly unfortunate. It is unfortunate because documents are a far more significant repository of humanity’s information. And documents can contain significantly richer information structures than data bases.

It is also unfortunate because the need for OIM of documents is greater now than ever. The demands of “repurposing” require that information be deliverable in multiple formats (paper-based, online, multimedia, hypermedia). And information must now be delivered through multiple channels (traditional bookstores and libraries, online services, the Internet).

Fortunately, in the past ten years a technology has emerged that extends to documents the data base’s capacity for data independence. And it does so without the data base’s restrictions on structural freedom. That technology is the “Standard Generalized Markup Language” (SGML), an official International Standard (ISO 8879) that has been adopted by the world’s largest producers of documents.

With SGML, organizations in government, aerospace, airlines, automotive, electronics, computers, and publishing (to name a few) have freed their documents from hostage relationships to processing software. SGML coexists with other data standards needed for OIM and acts as the framework that relates objects in the other formats to one another and to SGML documents.

As the enabling standard for OIM of documents, SGML necessarily plays a leading role in this series. We provide tutorials on SGML and other key standards and the techniques for applying them. Our books are not addressed solely to technical readers; we cover topics like the business justification for OIM and the business aspects of commerce in electronic information. We share the practical experience of organizations and individuals who have applied the techniques of OIM in environments ranging from immense industrial publishing projects to self-publishing on the World Wide Web.

Our authors are expert practitioners in their subject matter, not writers hired to cover a “hot” topic. They bring insight and understanding that can only come from real-world experience. Moreover, they practice what they preach about standardization. Their books share a common standards-based vocabulary. In this way, knowledge gained from one book in the series is directly

applicable when reading another, or the standards themselves. This is just one of the ways in which we strive for the utmost technical accuracy and consistency with the OIM standards.

And we also strive for a sense of excitement and fun. After all, the challenge of OIM—preserving information from the ravages of technology while exploiting its benefits—is one of the great intellectual adventures of our age. I’m sure you’ll find this series to be a knowledgeable and reliable guide on that adventure.

About the Series Editor

Dr. Charles F. Goldfarb is the inventor of SGML and HyTime, and technical leader of the committees that developed them into International Standards. He is an information management consultant based in Saratoga, CA.

About the Series Logo

The rebus is an ancient literary tradition, dating from 16th century Picardy, and is especially appropriate to a series involving fine distinctions between things and the words that describe them. For the logo, Andrew Goldfarb, who also designed the series’ “Intelligent Icons,” incorporated a rebus of the series name within a stylized SGML comment declaration.

The Charles F. Goldfarb Series on Open Information Management

**Turner, Douglass, and
Turner**

README.1ST: SGML for Writers and Editors

Adobe PageMaker, Adobe Illustrator, Adobe PostScript, FrameBuilder, Macintosh, Microsoft Word, Microsoft Windows, MS-DOS, NCSA Mosaic, Novell WordPerfect, SoftQuad Author/Editor, UNIX, VMS, Wordstar, and any other product names mentioned in this book are all trademarks or registered trademarks of their respective owners.

Series logo and "Intelligent Icons"™ by Andrew Goldfarb for EyeTech Graphics, © Andrew Goldfarb, 1995.

Series foreword and book foreword © Charles F. Goldfarb, 1995.

Excerpts from the following International Standards are © International Organization for Standardization, 1986, 1988, 1992, and are included with the kind permission of the copyright owner:

ISO 8879:1986, Information processing—Text and office systems— Standard Generalized Markup Language (SGML)

ISO/IEC 10744:1992, Information technology—Hypermedia/Time-based Structuring Language (HyTime)

Complete copies of these standards can be obtained from the national member body of ISO in your country, or contact: ISO, Case postale 56, CH-1211 Geneva 20, Switzerland.

The authors of this book have included a diskette of related materials as a convenience to the reader. The Series Editor did not participate in the preparation, testing, or review of the diskette and is not responsible for its contents.

FOREWORD

README.1ST: SGML for Writers and Editors is a first in more ways than just its title.

- It is the first book on SGML that was truly written for non-technical end users. That is undoubtedly because its authors—Ron Turner, Tim Douglass, and Audrey Turner, of Soph-Ware Associates—were professional writers and educators before they became SGML experts.
- It is the first SGML textbook, in the sense that it was developed in a university setting. It was proven in classroom use at Eastern Washington University, home of the “Electronic Information Institute” World Wide Web site.
- It is a beginner's book—easy and fun to read—but without the mistakes common to such books. That is, it doesn't try to achieve simplicity by blurring vital distinctions or omitting key concepts. Instead, it simplifies advanced SGML applications like hypertext and the World Wide Web by explaining, step-by-step, how they evolve naturally from basic SGML facilities.

In achieving all this, the book is scrupulously faithful to the SGML International Standard. It uses the standardized professional vocabulary of SGML to teach the language's constructs. But it introduces and motivates the use of that vocabulary by relating it to the real work of writers and editors. The title says *README.1ST*. I wish I could have. You'll be glad you can.

Charles F. Goldfarb
Saratoga, CA
June, 1995

PREFACE

This book grew out of dismay on the one hand and enthusiasm on the other. Like many who work with documents that must move among various platforms and systems, we were dismayed at the effort that we needed to spend repeatedly just to move one from here to there to do this and that. For us, moving documents meant learning every sort of transfer trick we could devise: downloading and uploading, exporting and importing, converting with third-party tools, scanning, and writing pieces of ad hoc conversion code to make it all work.

Our enthusiasm came from the promise of SGML. If SGML were indeed a standard that applied itself in a *generalized* manner, then it would let us spend more of our lives writing, editing, and producing documents and less time moving them around and recycling them.

But we were also dismayed at how difficult it seemed to know *where to begin* with SGML. True, the standard is in place, the tools are on the market, there is an active and helpful community on the Internet, and the various methodologies are working for a multitude of writers. But we could find nothing like a “One-Stop SGML” that would give the novice everything necessary to achieve SGML productivity. This book and its ancillary software were the result: a true introduction to the standard, a readable introduction to the essentials of SGML, and sufficient desktop software to enable the reader to get productive with SGML.

These are austere times for writers. Budgets for dedicated technical writing groups are shrinking or disappearing. Writing is being forced “upstream,” onto the desktops and benchtops of engineers, researchers, technicians and even secretaries. And this workplace cannot generally afford the luxury of week-long training sessions, on-site or at some hotel far away. Instead we must now engage in “just-in-time learning.” *README.1ST* is a learning tool for this new breed of publishers. We have attempted to address each new topic by starting in familiar territory and moving intuitively into a more formal discussion of that topic. The presentation is therefore more verbose than you yourself would be in writing technical documentation. Also, you will find several instances of repeated discussion on the same topic. This sort of review, while perhaps out of place in a reference document, helps the reader of a textbook to “nail down” troublesome and unfamiliar concepts.

A persistent concern for writers unaccustomed to formal standards is that a standard like SGML will somehow deprive them of their creativity. Our first and probably most important task therefore is to convince you that SGML enables and supports a writer's creativity. It does so by

allowing him or her to concentrate on creating text for a document without having to worry about how the document will appear.

How you decide to use *README.IST* depends both on your job description and on your learning style. If you are a manager faced with the decision of whether to commit to the SGML standard, then you should read at least the first three chapters. If you are a writer who wishes only to get the idea of how standard markup fits within the scheme of electronic documents, you should study carefully the first five chapters. Chapters six through eight cover the essential syntax of an SGML document type definition (DTD).

“But do I really need to know about a DTD, what with all the SGML editing tools currently on the market?” While we devote a good part of Chapter twelve answering that question, we suggest that being able to read a DTD intelligently is like having available the “Reveal Codes” feature of a popular word processor—it means that you can always see what is *really* going on in your document. And while we do not target this book to designers of document types, we progressively design and build a DTD as the core of our discussion over several chapters. Watching a DTD develop in step-wise fashion into the definition of a rather robust hypertext document proves to be a satisfying and effective method for understanding both the what and the why of a DTD.

Today there is enormous interest in HyperText Markup Language (HTML), the markup scheme for documents on the World-Wide Web. We offer a close-up look at HTML that lets you understand HTML as a particular application of SGML. This will provide you with an insight into HTML that goes far beyond the beginner's concerns over which tags to put where.

At this point in a preface authors typically lament the travails of having gotten a project finally into print. Yes, we neglected several contracting opportunities in our company, our family lives suffered, house plants died from lack of attention, pets did not get petted as they ought, dust gathered where it ought not, and all of the other things happened that we find in the litanies of most prefaces. But one such common phenomenon did *not* happen to us: burnout. We are no less excited about SGML and its benefits than when we began this project. We hope that *README.IST* succeeds in conveying that same excitement to you.

Acknowledgments

One tradition for a preface that we must not neglect is that of expressing thanks to those who have helped us along the way. The most important personage in the project has been Charles F. Goldfarb, who has patiently endured watching his child being mauled by pagans who, for purposes of this title, have frequently behaved more as practitioners and trainers than as theoreticians. While his name appears on the cover, and while his presence over our shoulders and in the series assures that what you read is *accurate*, we have aimed at creating a package that is also *readable* and *teachable*. If the book fails on those counts, we take all the blame.

Every vision requires an infrastructure to make it work. Mark Taub, editor at Prentice-Hall for the Charles F. Goldfarb series in the Professional Technical Reference Division, has assumed a unique professional profile that is well suited to the uniqueness of this series. To us he has func-

tioned as coach, cheerleader, dispatcher, comforter, and referee. To the extent that the series is successful, it will be due largely to Mark's accurate sense of the publisher's role in the evolving electronic document workplace and to his own role in steering the corporate ship through the murky waters of an evolving technology.

This has been a textbook example of an SGML project. The manuscript, prepared as SGML text and then converted and transmitted electronically to the publisher, enjoyed all the benefits of electronic publishing that we describe in the book itself. But for this to happen required knowledge, patience, and courage on the part of the publisher's production staff. Prentice-Hall Production Editor Camille Trentacoste has abundantly exhibited all of these virtues, and without her invaluable assistance this project would not have been completed. Camille also deserves high praise for her work on the style for this book and the entire Charles F. Goldfarb series.

We also thank Eliot Kimber, whose pre-publication version of his book on HyTime guided us in doing Chapter fourteen. Daniel W. Connolly of HaL Systems, the designated watchman of the turbulent world of HTML, offered criticism and correction that was most welcome.

We are grateful to James Clark, whose programming resulted in the updated public domain parser SGMLS (included on the disk which accompanies the text). We enhanced that code further to produce the imbedded validating parser which supports our browser (also on the disk).

We want also to thank Microsoft Corporation and Novell (WordPerfect) for their cooperation in supplying pre-release copies of their SGML authoring tools. SoftQuad provided software and support for the editing environment of the book as an SGML document. And Frame Technologies likewise provided software for the final composition of the camera-ready text.

At several critical junctures we relied on computers at the Spokane Intercollegiate Research and Technology Institute (SIRTI). Charles Aude, manager of the Computer Laboratory at SIRTI, helped make everything work the first time.

For her behind the scenes work in providing us with a very helpful review of the manuscript and for the assistance she provided on testing the software that accompanies *README.1ST*, we would like to thank Beth Breidenbach.

The other indispensable contributors to the project are the patient souls who suffered through an earlier draft of this book in the premier go-around of the Electronic Document course at Eastern Washington University. They deserve much, and the least we can do is to immortalize them here: Charli Anderson, Rhonda Bowers, Heather Brown, Lou Chobot, Kathryn Conlin, Emilee Dharmakul, Lou Dunham, Cheryl Fronk, Mark Galioto, Katherine Gooding, Mahilani Gutina, John Herman, Mary Jordan, Celeste Kaylor, Georgina Olsen, Penny Robinson, Brian Schauer, Terry Shatto, Cathy Towne, and Connie Woodard. May their text (and yours) live forever!

And in the spirit of the interactive smart document, we thank in advance those of our readers who will share their comments with us to benefit readers of future editions of this book.

Ronald C. Turner

Timothy A. Douglass

Audrey J. Turner

Soph-Ware Associates

swa@comtch.iea.com

CONTENTS

FOREWORD	XVII
PREFACE	XIX
1. WRITING WITH STANDARDS	1
<i>Yesterday, Today and Tomorrow</i>	
1.1 Why Are Standards Necessary?	2
1.2 Brief History and Survey of Standards	3
1.3 Objects of Standards	5
1.4 Impetus for an Electronic Standard.	6
<i>Automation</i>	
1.5 The Text Processing Application	7
1.6 Smart Document Architecture	10
2. SGML AS A NEW PARADIGM	13
<i>Structure before Style</i>	
2.1 Paradigms and Paradigm Shift	14
2.2 The Document.	15
<i>Printed Page versus Information to Be Processed</i>	
2.3 Writers and Editors	18
<i>Makers of Pages or Describers of Structures</i>	
2.4 Traditional Markup versus SGML Markup	18
2.5 The Format of the SGML Document	20
<i>Plain Unformatted Text</i>	
2.6 Example of SGML Markup	21
<i>List and List Items</i>	

3.	FEATURES AND BENEFITS OF SGML	27
3.1	Standards	28
	<i>Nice-to-Have versus Need-to-Have</i>	
3.2	Features and Benefits of the Lively Document	29
	Portability	30
	In-House Typesetting	31
	Word Processing	31
	System platforms	32
	Device Types	33
3.3	Precision of the Specification	33
3.4	A Wider Community	34
3.5	Familiar architectures	34
3.6	Open-Ended Document Processing	35
3.7	Flexibility of Markup	35
3.8	Standard Notation for Defining Document Types	36
3.9	Markup for Structure Only	36
4.	DOCUMENT ANALYSIS	39
	<i>Recognizing a Document's Structure</i>	
4.1	What Is Document Structure?	39
4.2	The Purpose of Structure	40
4.3	Indicating Structure Visually	41
4.4	Using SGML to Indicate Structure	44
	Document Structure	44
	<i>A Discussion</i>	
	Document Structure	47
	<i>Summary by Table</i>	
5.	DOCUMENT TYPE DEFINITION	51
	<i>Writing for Structure</i>	
5.1	Traditional Writing Manuals	52
5.2	Abstract of ANSI Z39.18-1987	52
5.3	ANSI Z39.18-1987 and Document Organization	53
5.4	The Deliverable in Electronic Authoring	55

5.5	Mechanics of Markup	55
	SGML Syntax for Tags	57
	Tags Enclose an Element	57
	Elements Can Be Nested	58
	Easing the Writer's Task	59
	<i>Markup Minimization</i>	
	Further Help	60
	<i>Validating SGML Parsers</i>	
5.6	The Ever-Present DTD	60
5.7	Document Type Definition	61
	<i>Basics</i>	
5.8	Document Type Definition	62
	<i>Working Discussion</i>	
	Allowable Element Types	63
	<i>Their Names and Their Meaning</i>	
	Tagging Communicates Structure	63
	Additional Properties of Element Types	64
5.9	Creating a Structured Document	65
	<i>A Case Study</i>	
	Description of the Document	65
	Structure of the Document	66
	Marked-Up Document	67
	Document as Printed	68
6.	ELEMENT TYPE DECLARATION	73
	<i>Building the Markup Vocabulary</i>	
6.1	Weaknesses of an Intuitive Tagging Scheme	74
6.2	Formal Tagging Convention for WAE	74
6.3	Formal Description of the Element Type's Usage	76
	Occurrence of an Element	76
	Contents of an Element	77
6.4	Formal Description of Order of Appearance	78
6.5	Recasting WAE as a Formal DTD	79
	Opening Delimiter to Name Parameter	79
	Omitted Tag Minimization Parameter	79
	Content Model Parameter	81
	<i>Sequence</i>	
	Content Model Parameter	82
	<i>Occurrence</i>	
6.6	Comments in the DTD	84

6.7	Document Type Declaration	86
6.8	Structure Definition	86
7.	ENTITIES	89
	<i>Making Text Reusable</i>	
7.1	Parameter Entities and General Entities	89
7.2	The Need for Parameter Entities	91
7.3	Defining and Using Parameter Entities	92
7.4	The Need for Character Entities	96
7.5	Declaring and Using Character Entities	96
7.6	Summary of DTD Symbols	97
7.7	Modified WAE DTD	98
8.	ATTRIBUTES	101
	<i>Enhancing the Power of Markup</i>	
8.1	Declaring and Using Markup	101
8.2	How Attributes “Fit” with Elements	103
8.3	How Attributes Are Used	105
	Figure or Table Identifications and References	106
	Indexable Terms	106
	Type of Signature Authority	106
	Security	107
	References to External Identification Schemes	107
8.4	Enhancing the WAE DTD with an Attribute	107
8.5	Markup for Attributes in a Workplace Document	111
9.	HYPERTEXT	113
	<i>Editing for Interactive Navigation</i>	
9.1	Linear versus Non-linear Text	114
9.2	Anticipating Non-Linear Navigation	116
9.3	Some Essential Terminology	117
9.4	Designing for Hypertext	118
	Explicit Links	118
	Implied Links	118
	“Value-Added” Links	118
9.5	Basic SGML Hypertext Linking Apparatus	119

9.6	Enhancing WAE for Hypertext Markup	120
	What Is Needed for a Hypertext Link	120
	An Element Type for Doing Link-To	121
	What That Element Type Accomplishes	121
	How We Mark the Anchor Element	122
9.7	Modifying WAE for Hypertext Linking	122
9.8	Implementation Issues	125
10.	ENTITY MANAGEMENT	127
	<i>Managing Reusable Text</i>	
10.1	Entities	127
	Formal Definition.	128
	Parameter Entities	129
	General Entities	130
10.2	Benefits	131
10.3	Entities in Pre-Electronic Writing and in SGML	131
10.4	Layered Activity	132
	Markup Representation	133
	Perceivable Representation	133
	Storage Representation	134
10.5	Lessons from Software Engineering.	135
10.6	Examples.	135
	A Document Type Definition in a Separate Entity	135
	One Logical Document With Several Physical Entities	136
	Details Isolated from the Body of the Text	139
11.	MARKED SECTIONS	143
	<i>Labeling for Special Purposes</i>	
11.1	Extending the Notions of Portability and Reusability	143
11.2	Declaring an SGML Marked Section	146
11.3	Syntax of Marked Sections.	146
11.4	Using Entities with Marked Sections	147
12.	READING A DTD	151
	<i>A Brief Walkthrough</i>	
12.1	DTD Fluency	151
	<i>Nice-to-Know or Need-to-Know?</i>	
12.2	Ground Rules for the Walkthrough.	155
12.3	A Short DocBook DTD Walkthrough	156