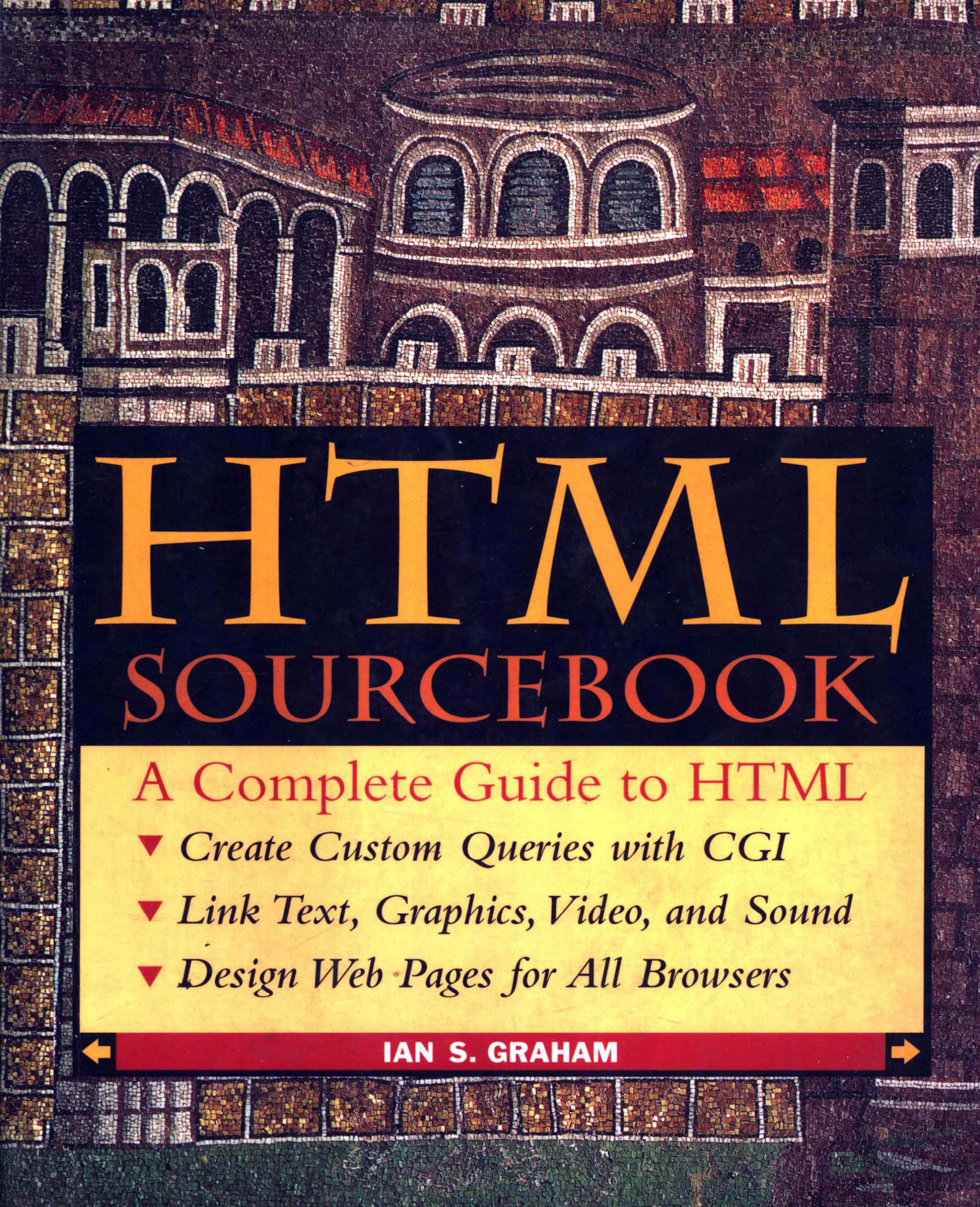

HTML SOURCEBOOK

A Complete Guide to HTML

- *Create Custom Queries with CGI*
- *Link Text, Graphics, Video, and Sound*
- *Design Web Pages for All Browsers*

IAN S. GRAHAM

THE HTML SOURCEBOOK

Ian S. Graham

John Wiley & Sons, Inc.

New York • Chichester • Brisbane • Toronto • Singapore

Publisher: Katherine Schowalter
Editor: Paul Farrell
Assistant Editor: Allison Roarty
Managing Editor: Frank Grazioli
Interior Design & Composition: Benchmark Productions, Inc.

This text is printed on acid-free paper.

Published by John Wiley & Sons, Inc.

Library of Congress Cataloging-in-Publication Data:
ISBN 0 471-11849-4

Printed in the United States of America

10 9 8 7 6 5 4

INTRODUCTION

It is fair to say that the World Wide Web project has taken the Internet by storm, confounding Internet skeptics and supporters alike. In hindsight, however, the reasons are obvious. The World Wide Web (WWW) model makes accessing the Internet easy, both to consumers of Internet-based information and to information providers. It's downright easy to distribute information via the Web, and just plain fun to go out and look for it. It is no surprise that World Wide Web utilities have grown, in less than three years, to be the most popular tools on the Internet.

A tool may be easy to use but often requires skill and training to be used well. This is certainly true of the tools involved in preparing and distributing information via hypertext documents and Internet hypertext servers. Preparing well-designed, useful, and reliable resources requires an in-depth understanding of how the tools that deliver these resources work and how to use them *well*. The intention of this book is to provide this understanding. Assuming that you are familiar with traditional Internet resources, such as FTP, telnet, electronic mail, and Gopher, there are essentially three new components to consider:

1. *Uniform Resource Locators*, or *URLs*, which are the scheme by which Internet resources are addressed in the WWW.
2. The *HyperText Transfer Protocol (HTTP)* and HTTP *client-server* interactions. HTTP servers are designed specifically to distribute hypertext documents, and you must know how they work to take advantage of their powerful features.
3. The *HyperText Markup Language*, or *HTML*. This is the markup language with which World Wide Web hypertext documents are written, and is what allows you to create hypertext links, fill-in forms, and clickable images. Writing good HTML documents involves both technical issues (proper construction of the HTML document) and design issues (ensuring that the information content is clearly presented to the user).

The goal of this book is to explain all three of these issues and to give you the tools to develop your own high-quality World Wide Web products. The remainder of this introduction looks briefly at these three components and explains their basic features. This is followed by an outline of the book and some suggestions as to how to best approach the text and examples.

UNIFORM RESOURCE LOCATORS

Uniform Resource Locators, or URLs, are a naming scheme for specifying how and where to find any Internet server resource, such as from Gopher, FTP or WAIS servers. For example, the URL that references the file ***macweb.zip*** in the directory ***/pub/web/browsers*** on the anonymous FTP server ***ftp.bozo.net*** is simply:

```
ftp://ftp.bozo.net/pub/web/browsers/macweb.zip
```

WWW hypertext documents use URLs to reference other hypertext resources.

THE HYPERTEXT TRANSFER PROTOCOL

The *HyperText Transfer Protocol*, or HTTP, is a new Internet protocol designed expressly for the rapid distribution of hypertext documents. Like other Internet tools, such as FTP, WAIS, or Gopher, HTTP is a *client-server* protocol. In the client-server model, a *client* program running on the user's machine sends a message requesting service to a *server* program running on another machine on the Internet. The server responds to the request by sending a message back to the client. In exchanging these messages, the client and server use a well-understood *protocol*. FTP, WAIS, and Gopher are other examples of Internet client-server protocols, all of which are accessible to a World Wide Web browser. However, the HTTP protocol is designed expressly for hypertext document delivery, so most of your communication will be with HTTP servers.

At the simplest level, HTTP servers act much like anonymous FTP servers, delivering files when clients request them. However, HTTP servers support additional important features:

- The ability to return to the client not just files, but also information generated by programs running on the server
- The ability to take data sent from the client and pass this information on to other programs on the server for further processing

These special server-side programs are called *gateway* programs, because they usually act as a gateway between the HTTP server and other local resources, such as databases. Just as an FTP server can access many files, an HTTP server can access many different gateway programs; in both cases, you can specify which resource (file or program) you want through a URL.

The interaction between the server and these gateway programs is governed by the *Common Gateway Interface (CGI)* specifications. Using the CGI specifications, a programmer can easily write simple programs or scripts to process user queries, interrogate databases, make images that respond to mouse clicks, and so on.

THE HYPERTEXT MARKUP LANGUAGE

The *HyperText Markup Language*, or *HTML*, is the language used to prepare hypertext documents. These are the documents you distribute on the World Wide Web and are what your human clients actually see. HTML contains commands, called *tags*, to mark text as headings, paragraphs, lists, quotations, emphasized, and so on. It also has tags for including images within the documents, for including fill-in forms that accept user input, and, most importantly, for including hypertext links connecting the document being read to other documents or Internet resources, such as WAIS databases and anonymous FTP sites. It is this last feature that allows you to click on a string of highlighted text and access a new document, an image, or a movie file from a computer thousands of miles away. And how does the HTML document specify where this document is? Through a URL, which is included in the HTML markup instructions and which is used by your browser to find the designated resource.

What resources can URLs point to? They can be other HTML documents, pictures, sound files, movie files, or even database search engines. They can be on your computer or anywhere on the Internet. They can be accessed from HTTP servers or from FTP, Gopher, WAIS, or other servers. The URL is an immensely flexible scheme and, in combination with HTML, yields an incredibly powerful package for preparing a web of hypertext documents linked to each other and to Internet resources around the world. This image of interlinked resources is, in fact, the vision that gave rise to the name World Wide Web.

OVERVIEW OF THE BOOK

This book is an introduction to HTML, URLs, HTTP, the CGI interface, and the design and preparation of resources for delivery via the World Wide Web. It begins with the HTML language. Almost every resource that you prepare will be presented through an HTML document so that your HTML presentation is your *face* to the world. It is crucial that you know how to write proper HTML, and that you understand the design issues involved in creating good documents, if you are to make a lasting impression on your audience and present your information clearly and concisely. It won't matter if the Internet resources you make available are the best in the world if your presentation of them is badly designed, frustratingly slow, or difficult to follow.

HTML is also an obvious place to start. You can write simple HTML documents and view them with a WWW browser, such as **Mosaic, MacWeb, lynx, Cello,** or **Netscape** without having to worry about CGI programs, HTTP servers, and other advanced features. You can also easily add to your documents URLs pointing to server resources around the world, and get used to how the system works: Browsers understand HTML *hypertext anchors* and the URLs they contain, and they have built-in software to talk to Internet servers using the proper protocols. You can accomplish a lot just by creating a few pages of HTML.

Chapter 1 is an introduction to HTML and to the design issues involved in preparing HTML documents. This nontechnical chapter combines a brief overview of HTML with important aspects of the document design process. The details of the HTML language and more sophisticated client-server issues are left to Chapters 2 and 3.

Design issues are very important in developing good World Wide Web presentations. HTML documents are not like text documents or traditional hypertext presentations, since they are limited by the varied capabilities of browsers and by the speed with which documents can be transported across the Internet. Chapter 1 discusses what this means in practice and gives guidelines for avoiding major HTML authoring mistakes. In most cases, this is done using examples with the important issues being presented in point form so that you can easily extract the main points on first reading.

One point that is emphasized in Chapter 1 and everywhere in the book is the importance of using correct HTML markup constructions when you create your HTML documents. Although HTML is a relatively straightforward language, there are many important rules specifying where tags can be placed. Ensuring that your documents obey these rules is the only way you can guarantee that they will be properly displayed on the many different browsers your clients may use. All too often, writers prepare documents that look wonderful on one browser, but end up looking horrible or even unviewable on others.

Although some general rules for constructing valid HTML are included in Chapter 1, Chapter 2 and the references therein should be used as detailed guides to correct HTML. In particular, Chapter 2 presents a detailed exposition of the HTML language and of the allowed nesting of the different HTML markup instructions. It also explains the syntax and rules for constructing URLs.

Chapter 3 explains the structure and syntax of Uniform Resource Locators (URLs). This is the addressing scheme used in HTML documents to indicate the target of hypertext links.

Of course, HTML is only a beginning. To truly take advantage of the system, you must understand the interaction between WWW client browsers and HTTP servers, and be able to write server-side gateway programs that take advantage of this interaction. Chapter 4 delves into the details of the interaction between WWW clients and HTTP servers and explains the Common Gateway Interface (CGI) specification for writing server-side programs that interface with the HTTP server. Chapter 4 includes simple examples to demonstrate the HTTP protocol and the CGI interface, as well as useful references to sites on the Internet that contain instructional interactive documents. Chapters 2, 3 and 4 are the technical core of this book and should be of use as reference material when writing HTML documents or server CGI programs.

Needless to say, there already are many useful CGI programs available on the Internet, ranging from the **imagemap** program for handling clickable images to sophisticated front-end packages for databases. These and other useful CGI tools are discussed in Chapter 5, which also looks at auxiliary tools useful in developing and organizing HTML documents. For example, there are tools for converting collections of e-mail letters into hypertext archives, or for creating a hypertext "Table of Contents" for large collections of related HTML files. The second half of Chapter 5 discusses these tools and indicates sites where they can be obtained. Almost all of them are available over the Internet, either from anonymous FTP sites or from HTTP servers. URLs are used to indicate the locations of these programs and of additional documentation when available.

Preparing HTML documents can be tedious, since HTML markup tags are complicated text strings that must be included in your text document. HTML codes are not only time consuming to type, but also a common source of error: It is easy to make a mistake typing all those tags! You will not be the first to notice this fact, and many individuals, groups, and com-

panies are actively developing HTML editors to help in the document creation process. Chapter 6 summarizes and briefly describes the various HTML editors available on PC, Macintosh, and UNIX platforms, and explains how to obtain them.

In some cases, you may not want an editor, but would rather be able to convert documents from another format, such as Microsoft's Rich Text Format, FrameMaker's MIF format, LaTeX, and so on into HTML. Chapter 6 includes a summary of several of these packages, including instructions on how they can be obtained. Finally, there are a number or useful tools for *validating* the HTML document for conformity with the language specification and for checking the validity of hypertext links within a document. These tools are listed at the end of Chapter 6.

Chapter 7 is a brief review of the different browsers available for exploring the WWW and for viewing HTML documents. You *should* be interested in how the documents you design will look on browsers other than the one you regularly use. Designing for a single browser is dangerous, since there is a tendency to tailor the HTML for the peculiarities of that particular program. This can lead to HTML documents that look fine on your browser, but horrible on others. It is wise to pick up another browser or two, just to avoid these problems.

Chapter 8 discusses the issues involved in setting up an HTTP server and briefly reviews the commonly available server packages. Again there are many URL references to additional documentation and to locations where server software can be obtained.

Chapter 9, the final chapter in the book, gives some examples of WWW sites containing interesting and well-designed presentations. These sites range from business and entertainment to those devoted to scientific research and education. I asked the creators of these sites to describe the creation process so that you can get a feel for what was involved. I encourage you to visit these sites and see how they work. I can guarantee you will be impressed.

Finally, you should just go out and browse! Writing a book and spouting off one's own ideas of good and bad design is all well and good, but you, as a writer of HTML documents, will appreciate how things look and feel only by going out there and looking and feeling. This book is merely a framework for appreciating what tens of thousands of creative individuals are already doing. So, go and see for yourself!

ACKNOWLEDGMENTS

I would first like to thank my coworkers in the Instructional and Research Computing Group at the University of Toronto: Without their support and encouragement this book would not have been possible. In particular I want to thank John Bradley and Anna Pezacki for giving me the time off to write, probably against their better judgement! I also must especially thank Allen Forsyth and Norman Wilson for critically reading several sections of the manuscript and for removing the "lumpiness" from my early drafts (thank you, Norman, for that elegant image). I also greatly benefitted from the technical expertise of Rudy Ziegler and Terry Jones, who helped me through several problems with image file conversion and computer networking.

Most importantly, I would like to thank my wife, Ann Dean, both for her editing skills and for her unwavering support and patience during the past few months. Without her, this book would not have been possible.

DEDICATION

To Ann

CONTENTS