

# MEDIAWARE REVIEW

## *A Critical Examination of Two Resources for Creating HTML Documents for the World Wide Web*

L. F. (Len) Proctor, Editor

---

*HTML Sourcebook: A Complete Guide to HTML* by Ian S. Graham. Published by John Wiley & Sons, Toronto, 1995, 416 pages, \$41.95.

*HTML Web Weaver* by Robert Best (Robert.Best@Potsdam.edu). For the latest release and pricing information you can contact:  
<http://www.potsdam.edu/Web.Weaver/About.htm>

---

*Reviewed by Richard Schwier*

This review examines two resources currently available for creating HTML documents for the World Wide Web, not because they are unusual, but rather because they are good examples of a rapidly growing pool of similar resources, and both point in the direction these types of resources are taking. One is a book; one is a text editor. One is Canadian; one is inexpensive. Both are comparatively good resources.

The Canadian book, by Ian Graham from the Instructional and Research Computing Group at the University of Toronto, addresses a very real and growing need. Most anyone trying to write documents for the World Wide Web realizes that although the initial commands are easy to learn and use, difficulty lurks just beneath the surface. I have hungered for a book about creating HTML documents that is easy to read and full of examples, and this book goes a long way in the right direction.

The book appears to be thorough, at least it went well beyond what I was looking for in terms of content. Its primary focus is on the HyperText Markup Language (HTML), but it also considers URLs (Uniform Resource Locators), HTTP (HyperText Transfer Protocol) and CGI (Common Gateway Interface). In all, it is a comprehensive treatment of things you will need

to consider to prepare resources and mount them on the World Wide Web. In fact, it includes much more than we commoners require to produce web documents.

Permit me to define a couple of terms so the uninitiated can still interpret most of this review. HTML is the major language used to create documents for the World Wide Web. It allows anyone to create documents on most any platform, and make them available to most everyone else. When you view documents on the World Wide Web, you use a program called a browser. There are many different browsers, but some common ones are Mosaic, Netscape and Lynx. These programs can read HTML documents (and other things) on the web.

First of all, for anyone who has not yet encountered the bittersweet characteristics of HTML, it is a **language** for formatting text on a screen. It is a *MIME* content type EM and it employs embedded commands in the text to orchestrate the presentation of elements.

For instance, this paragraph, if saved as an HTML document, would be interpreted by my Netscape browser as:

In the first three chapters, Graham introduces the reader to HTML and some technical design issues. Chapter 1 gives general rules and ideas for constructing HTML documents, and Chapter 2 wades in with more detail. Chapter 3 explains how to construct and use URLs, the method used in HTML documents to address documents and make hypertext links. Graham is careful to emphasize that design is critical, and he quite properly nags the reader to use proper HTML constructions and syntax when creating HTML documents. In just my first reading, I picked up a handful of ideas for correcting mistakes I'd inadvertently made in files. For instance, Graham informs those writing HTML documents that HTML recognizes only the largest spacing value of a logical structure element, What? This means that if one uses several paragraph tags `<P>` to create some space between items on a page, only the first will be invoked, and the rest ignored. Thus, `<P><P><P><P><P>` renders the same spacing on the page as `<P>`. Graham also reminds the reader that logical highlighting elements should be used wherever possible so that various users can set the preferences of their browsers to impose the styles they want on the page. For example, page designers can be tempted to bold-face a word or phrase by placing the element between `<B>` and `</B>` in the text. This will force the browser to display the text as bold, and it is called a physical highlighting element. Alternatively, the page designer can use `<STRONG>` to identify the highlighted text, and the browser will use whatever convention the user prefers to display text

with strong emphasis. I might set my browser to display as red italics rather than as bold faced text.

This is precisely the point at which instructional designers flinch. Give the user control over document design? Has the world gone mad? With HTML documents, users are given enormous control over the “look” of documents they view, and a designer who wishes to wrest that control away from the user must go to extreme lengths. Thus, the point needs to be made that HTML is not, first and foremost, an instructional design tool—nor is it an authoring language for CBI. It is simply a protocol for structuring information and providing Hypertext links within and between documents. As a design tool, it is not as powerful as many other programs, and as an authoring language, it pales next to most any other available program. But it is a wonderful tool for structuring information, it is truly a cross-platform tool, and everyone (well almost everyone) is using it.

And this is where Graham’s book shines. It acknowledges, but ignores, the platform debate and focuses directly on using the tool properly. The book is filled with examples, and I found this to be one of the best features of the book. In most of the HTML references I have seen (most downloaded from web sites), I am left with the impression that I joined half of a conversation in progress. While the references are useful, there seem to be missing instructions or assumptions about the reader’s knowledge. Graham clears up much of the confusion with clear examples of HTML constructions, and he wisely follows them with illustrations of how the constructions might be rendered by different browsers. Let me publicly say, “Thank you, Ian Graham.”

There are a host of other good tips and reminders. At the same time, the reader must remember to take some of the advice with a grain of salt. For example, Graham admonishes:

“You also don’t want the link [highlighted text] to be gratuitously long. It is much more effective to link a single word or selection of words than to a whole sentence.” (p.29)

Now, while I would agree with the intention of the statement, and I would probably try to follow the advice, there is no evidence I know which suggests that words and phrases provide more effective links than sentences for the user. There may be times when complete sentences or even paragraphs might be effectively linked to another document, although I can think of few cases where it would be advisable. Similarly,

“Although you can build hypertext links within a document to other points inside the same document, this is generally more difficult to navigate than a collection of smaller files.” (p.44)

I'm not sure what exactly is meant by "more difficult to navigate" but links appear the same regardless of the document structure, so I doubt he means that one is easier to use than another. Perhaps Graham is suggesting that movement within a large document is slower than between documents, and certainly it is true that larger documents can take a long time to transfer from the host to your machine. It is also consistent with his advice to keep documents and image file sizes small. But this is not consistent with my own experience when remote access is initiated each time a new tile is requested from a very busy site. Performance can be sluggish, and sometimes a request is even refused by the host. In these cases, I might prefer links within a document I've already accessed.

These are not severe criticisms of Graham's advice, but rather a reminder to instructional designers who use this book that it emphasizes how to write effective HTML documents-not effective instructional documents, and some of the advice may be at odds with good instructional design.

In short, the first three chapters of this book made it worth having, and I recommend it for anyone wanting to create HTML documents. Turning to the rest of the book, Chapter 4 discusses the HTTP protocol for delivering hypertext materials and the Common Gateway Interface (CGI) specification which defines the way a server communicates with gateway programs. This is very illuminating and important information if you are setting up your own server, or have occasion to communicate with people operating servers at a web site.

Chapter 5 discusses HTML and CGI tools, and it is particularly useful if you want to do anything beyond basic HTML document development. For example, we have been struggling with how to use a single graphic in our designs and make various sections of the graphic "hot" to establish links with other documents. Sounds simple, doesn't it? Well, it is a fairly convoluted process which requires additional tools and some technical expertise. This and other operations require special utility programs and CGI programs to create and execute the commands, and it is one of the places where mere mortals get a little weak in the knees. Graham's style is very inviting in this chapter, but the information is still somewhat daunting for most readers.

Chapter 6 briefly describes several editors available for PC, Macintosh and UNIX platforms, complete with ordering information. It also lists a number of document translation packages which allow you to create documents with a word processor and then either translate them or save them as HTML documents. This chapter is helpful, but it will certainly suffer the ravages of a changing marketplace very quickly. There are many new editors available since the publication of this book, and there will most certainly be many more available by the time this review is published. One thing we can look forward to is the inclusion of HTML as a "save as" option in all major word processing packages. Extensions are already available for many of

them. A similar fate awaits Chapter 7, which reviews several of the different browsers available for exploring the web. The web seems to be solidifying, or at least curdling, around one or two browsers. In fact, the wildly popular "Netscape" is mentioned only as a coming attraction, demonstrating just how volatile information in this field can be. Still, Graham's discussion of cross-browser design contains good advice that will certainly transcend immediate changes in the marketplace.

Chapter 8 describes HTTP servers and server utilities, and it is important reading for anyone setting up a web site. It considers some of the complications one can encounter, and does a credible job of comparing some of the advantages of some platforms over others as servers.

Chapter 9 is a clear departure from the rest of the book, and it is one of the book's most engaging sections. Graham chose a handful of exemplary web sites and asked the creators of the sites to write descriptions of the important issues and design challenges they faced. This chapter displays some of the diverse and creative issues one faces in site development, and I found the descriptions to be accessible and illustrative.

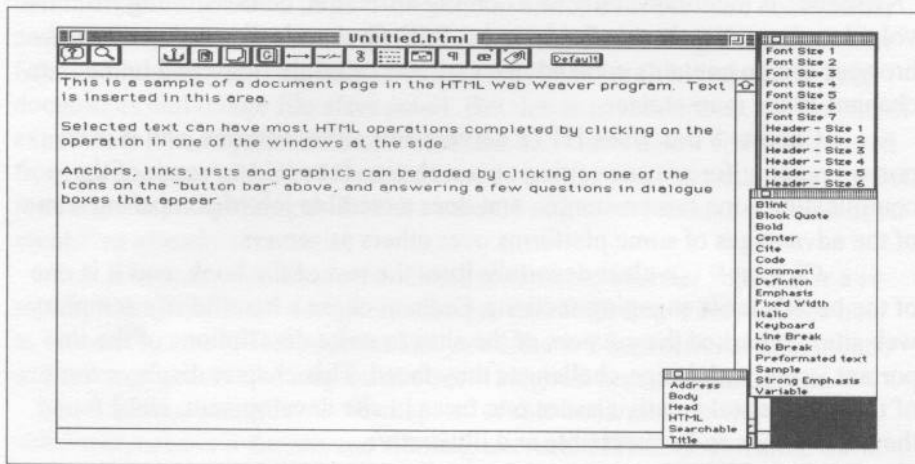
Overall, I found this work to be among the best I have yet encountered. Certainly this is a mushrooming area, and there will no doubt be other excellent resources available in short order. I found that much of the technical information in the middle chapters to be either beyond my knowledge level, or well beyond the scope of my interest. I'm confident, however, that others will find these chapters more useful than the ones I singled out for praise. Regardless, this is a good reference, and I think Graham can keep very busy in the next several years writing new editions.

### *An HTML Document Editor*

As outlined in Graham's book, a large and growing number of editors are available. Think of these editors as word processors specifically designed for creating HTML documents. Editors are vitally important tools for creating HTML documents, because they remove some of the drudgery and complexity from the process, thus making it much easier for individuals who have little acquaintance with programming protocols to create documents. Generally speaking the role of an HTML editor is to take text and add the HTML markup tags and instructions.

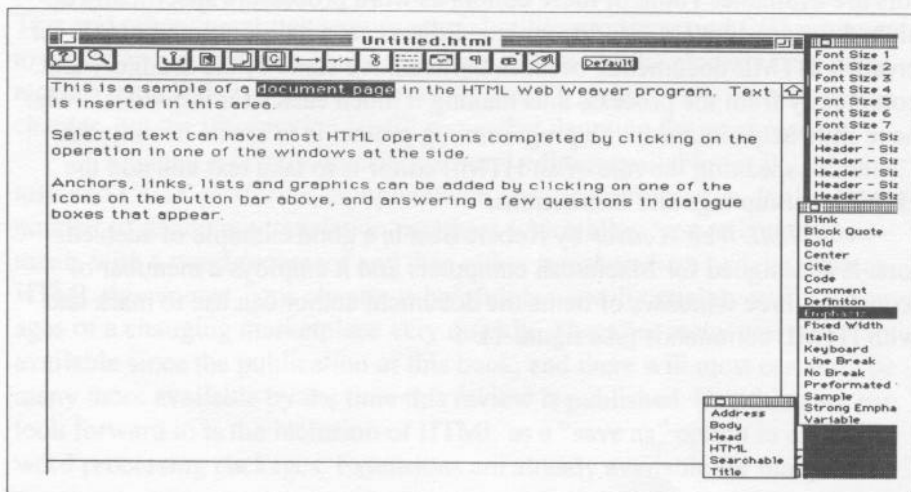
*HTML Web Weaver* by Robert Best is a good example of such editors. It is designed for Macintosh computers and it employs a \_\_\_\_\_ of icons and three windows of items the document author can use to mark text with HTML commands (see figure I).

Figure 1  
Main document construction page from HTML Web Weaver, including menu buttons and command windows.



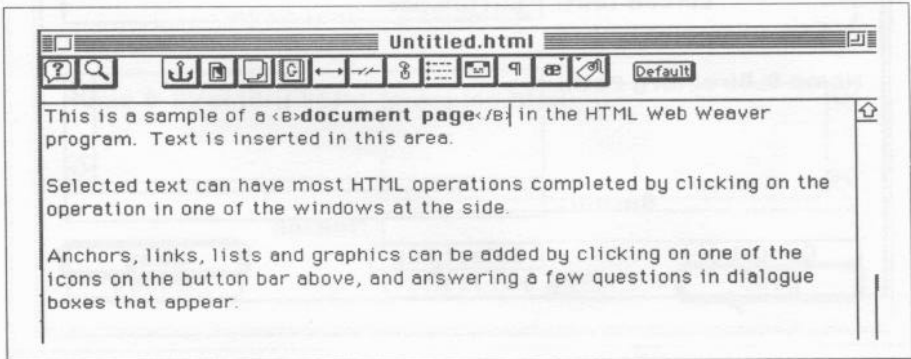
To give an idea of how *HTML Web Weaver* can be used, a few simple examples will be used. First, text can be inserted, constructed, pasted or imported into the document. Once on the screen, text is selected and then the desired HTML function selected from one of the windows. For this example, the text "document page" is selected and the HTML command for emphasizing the phrase is chosen (see Figure 2).

Figure 2  
Selection of text and associated emphasis command from HTML Web Weaver windows.



This action causes the insertion of the emphasis commands within the text of the page, and surrounding the selected text, in this case **document page B** (see Figure 3). Interestingly, *HTML Web Weaver* inserts the physical command for bold facing the text rather than the logical command for emphasis which would employ whatever setting the user chooses.

Figure 3  
*HTML document with emphasis command.*



One of the tricky procedures required in HTML documents is linking an item to another document, a procedure at the heart of hypertext, and one which requires a series of precise commands. *HTML Web Weaver* makes the process quite easy. First, text is selected which will act as the link to another document. Then the button which looks like chain links is clicked. A dialogue box appears, and the user clicks the "Select File" button and then selects the appropriate file which will be the destination when the previous text is clicked (see Figures 4, 5, 6 and 7).

Figure 4  
*Highlighting text and selecting the "link button".*

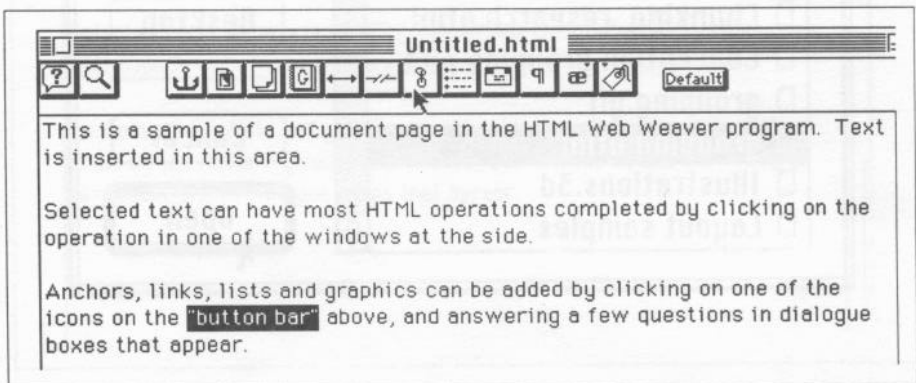


Figure 5  
Choosing the "Select File" button from Link Editor dialogue window.

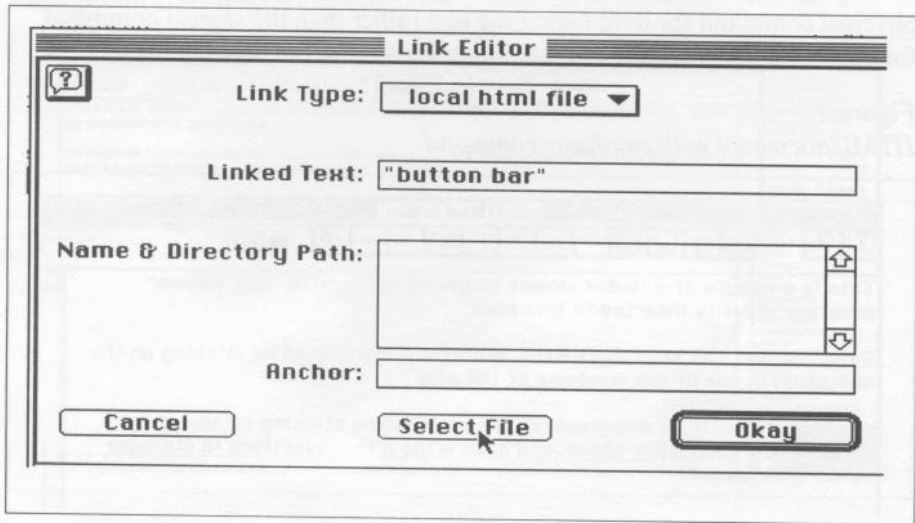


Figure 6  
Selecting the destination file from dialogue box.

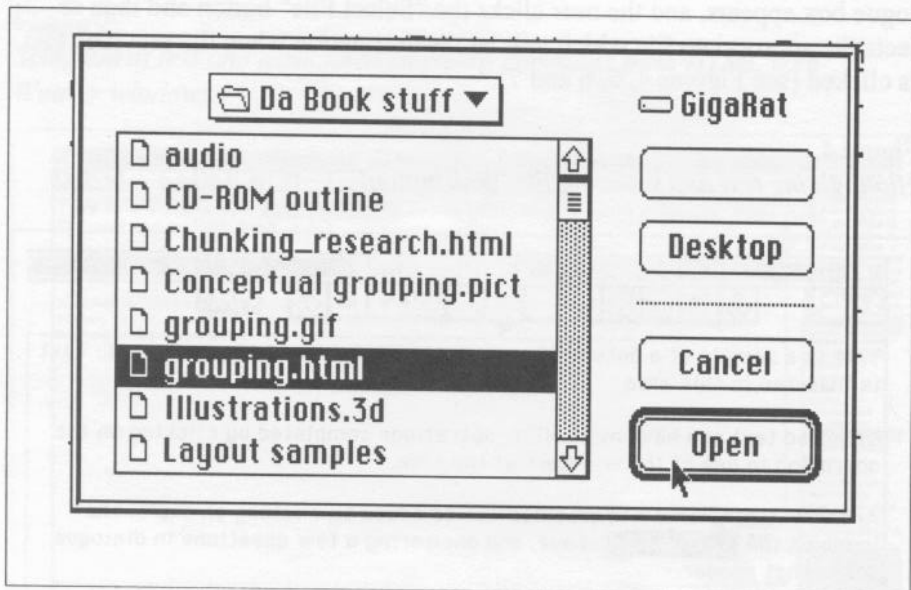




Figure 7  
 Identifying the destination file in Link Editor dialogue window.

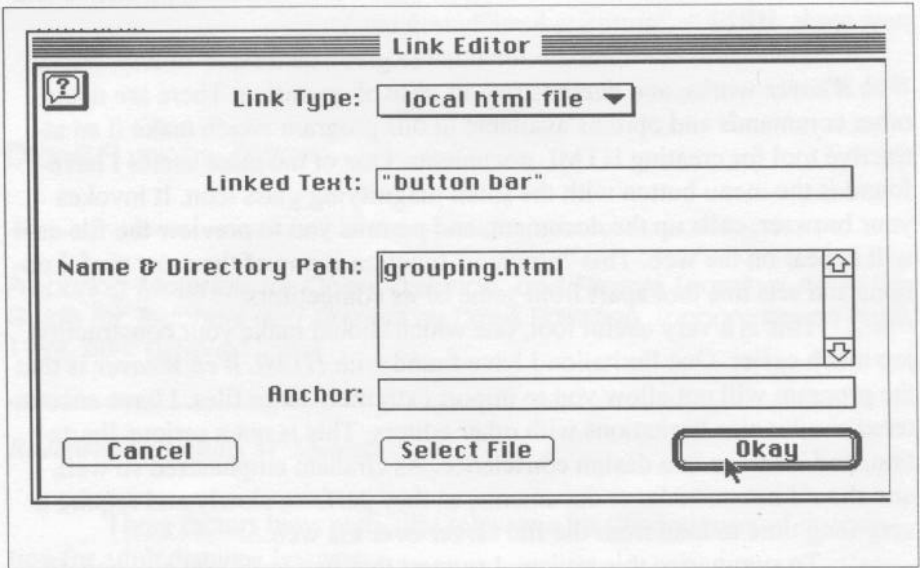
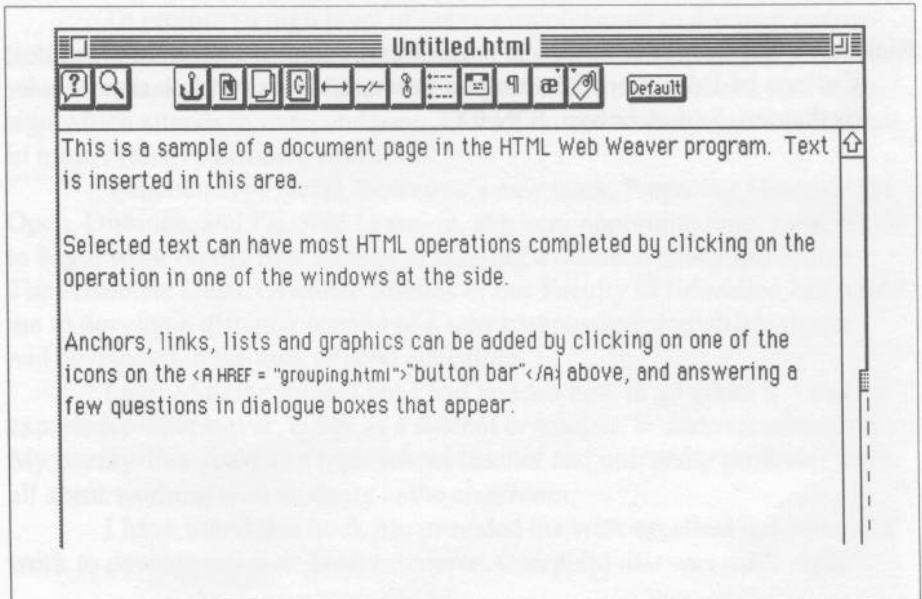


Figure 8  
 Linking command appearing in the document window.



In precisely the same manner as earlier demonstrated, the HTML command required to link the text to the selected file is inserted automatically into the text of the document (see Figure 7). In this case, the active text now reads `HREF = "grouping.html"button barA`.

These few examples are intended to give you a sense of how HTML *Web Weaver* works, and demonstrate its ease of operation. There are many other commands and options available in this program which make it an attractive tool for creating HTML documents. One of the most useful I have found is the menu button with the small magnifying glass icon. It invokes your browser, calls up the document, and permits you to preview the file as it will appear on the web. This "preview" function is one of the most useful options and sets this tool apart from some of its competitors.

This is a very useful tool, one which should make your construction job much easier. One limitation I have found with *HTML Web Weaver* is that the program will not allow you to import extremely large files. I have encountered similar size limitations with other editors. This is not a serious limitation, and it serves as a design conscience. As Graham emphasized so well, one should not make large documents, as they perform slowly and require a very long time to load from the file server over the web.

To summarize this review, I suggest that two tools you will benefit greatly from as you begin to design your own HTML documents are a good HTML reference book and a good HTML editor. These two examples, while probably not the only resources I would characterize as good, are certainly choices I would recommend as you begin your journey and I continue mine.

---

#### REVIEWER

Richard A. Schwier is a professor, in the Department of Curriculum Studies, College of Education, 28 Campus Drive, University of Saskatchewan, Saskatoon, Saskatchewan, S7N 0X1.