

# **HTMLDOC 1.8.27 Software Users Manual**

ESP-003-20060801

Easy Software Products  
Copyright 1997-2006, All Rights Reserved.



# Table of Contents

<b><u>Introduction</u></b>	<b>IN-1</b>
<u>History</u>	IN-1
<u>Organization of This Manual</u>	IN-2
<u>Support</u>	IN-2
<u>Encryption Support</u>	IN-2
<u>Copyright, Trademark, and License Information</u>	IN-2
<b><u>Chapter 1 - Installing HTMLDOC</u></b>	<b>1-1</b>
<u>Requirements</u>	1-1
<u>Installing HTMLDOC</u>	1-1
<u>Installing HTMLDOC on Microsoft Windows</u>	1-1
<u>Installing HTMLDOC on MacOS X</u>	1-2
<u>Installing HTMLDOC on Linux</u>	1-3
<u>Installing HTMLDOC on Solaris</u>	1-3
<u>Licensing HTMLDOC</u>	1-3
<u>Uninstalling HTMLDOC</u>	1-4
<u>Uninstalling HTMLDOC on Microsoft Windows</u>	1-4
<u>Uninstalling HTMLDOC on MacOS X</u>	1-4
<u>Uninstalling HTMLDOC on Linux</u>	1-4
<u>Uninstalling HTMLDOC on Solaris</u>	1-4
<b><u>Chapter 2 - Getting Started</u></b>	<b>2-1</b>
<u>Starting HTMLDOC</u>	2-1
<u>Choosing a HTML File</u>	2-2
<u>Setting the Output File</u>	2-3
<u>Generating the Document</u>	2-4
<b><u>Chapter 3 - Generating Books</u></b>	<b>3-1</b>
<u>Overview</u>	3-1
<u>Choosing HTML Files</u>	3-2
<u>Selecting a Title File</u>	3-2
<u>Setting the Output Format</u>	3-3
<u>Setting the Output File</u>	3-3
<u>Generating the Document</u>	3-3
<u>Saving Your Book</u>	3-3
<b><u>Chapter 4 - HTMLDOC from the Command-Line</u></b>	<b>4-1</b>
<u>Getting to the Command-Line on Windows</u>	4-1
<u>The Basics of Command-Line Access</u>	4-2
<u>What Are All These Commands?</u>	4-2
<u>Converting Multiple HTML Files</u>	4-3
<u>Generating Books</u>	4-3
<u>What are all these commands?</u>	4-3
<u>Setting the Title File</u>	4-4
<u>Putting It All Together</u>	4-4
<b><u>Chapter 5 - Using HTMLDOC on a Web Server</u></b>	<b>5-1</b>
<u>The Basics</u>	5-1
<u>Using HTMLDOC as a CGI Program</u>	5-1
<u>Server-Side Preferences</u>	5-2
<u>Configuring HTMLDOC</u>	

# Table of Contents

## **Chapter 5 - Using HTMLDOC on a Web Server**

<u>Using HTMLDOC From Server-Side Scripts and Programs</u> .....	5-8
<u>Calling HTMLDOC from a Shell Script</u> .....	5-9
<u>Calling HTMLDOC from Perl</u> .....	5-10
<u>Calling HTMLDOC from PHP</u> .....	5-10
<u>Calling HTMLDOC from C</u> .....	5-12
<u>Calling HTMLDOC from Java</u> .....	5-13

## **Chapter 6 - HTML Reference**.....6-1

<u>General Usage</u> .....	6-1
<u>Elements</u> .....	6-2
<u>Comments</u> .....	6-4
<u>Header/Footer Strings</u> .....	6-6
<u>FONT Attributes</u> .....	6-7
<u>Headings</u> .....	6-7
<u>Numbered Headings</u> .....	6-8
<u>Images</u> .....	6-8
<u>Links</u> .....	6-8
<u>META Attributes</u> .....	6-9
<u>Page Breaks</u> .....	6-9
<u>Tables</u> .....	6-9

## **Chapter 7 - GUI Reference**.....7-1

<u>The HTMLDOC GUI</u> .....	7-1
<u>Document File Operations</u> .....	7-1
<u>New</u> .....	7-1
<u>Open</u> .....	7-1
<u>Save</u> .....	7-2
<u>Save As</u> .....	7-2
<u>Generate</u> .....	7-2
<u>Close</u> .....	7-2
<u>The Input Tab</u> .....	7-3
<u>Document Type</u> .....	7-3
<u>Input Files</u> .....	7-3
<u>Add Files</u> .....	7-3
<u>Edit Files</u> .....	7-3
<u>Delete Files</u> .....	7-4
<u>Move Up</u> .....	7-4
<u>Move Down</u> .....	7-4
<u>Logo Image</u> .....	7-4
<u>Title File/Image</u> .....	7-4
<u>The Output Tab</u> .....	7-5
<u>Output To</u> .....	7-5
<u>Output Path</u> .....	7-5

# Table of Contents

## **Chapter 7 - GUI Reference**

<u>Header and Footer</u> .....	7-8
<u>The TOC Tab</u> .....	7-9
<u>Table of Contents</u> .....	7-9
<u>Numbered Headings</u> .....	7-9
<u>Header and Footer</u> .....	7-9
<u>Title</u> .....	7-9
<u>The Colors Tab</u> .....	7-10
<u>Body Color</u> .....	7-10
<u>Body Image</u> .....	7-10
<u>Text Color</u> .....	7-10
<u>Link Color</u> .....	7-10
<u>Link Style</u> .....	7-10
<u>The Fonts Tab</u> .....	7-11
<u>Base Font Size</u> .....	7-11
<u>Line Spacing</u> .....	7-11
<u>Body Typeface</u> .....	7-11
<u>Heading Typeface</u> .....	7-11
<u>Header/Footer Size</u> .....	7-12
<u>Header/Footer Font</u> .....	7-12
<u>Character Set</u> .....	7-12
<u>Options</u> .....	7-12
<u>The PS Tab</u> .....	7-12
<u>PostScript Level</u> .....	7-12
<u>Send Printer Commands</u> .....	7-13
<u>Include Xerox Job Comments</u> .....	7-13
<u>The PDF Tab</u> .....	7-13
<u>PDF Version</u> .....	7-13
<u>Page Mode</u> .....	7-14
<u>Page Layout</u> .....	7-14
<u>First Page</u> .....	7-14
<u>Page Effect</u> .....	7-14
<u>Page Duration</u> .....	7-14
<u>Effect Duration</u> .....	7-14
<u>The Security Tab</u> .....	7-15
<u>Encryption</u> .....	7-15
<u>Permissions</u> .....	7-15
<u>Owner Password</u> .....	7-15
<u>Options</u> .....	7-15
<u>User Password</u> .....	7-16
<u>The Options Tab</u> .....	7-16
<u>HTML Editor</u> .....	7-16
<u>Browser Width</u> .....	7-16
<u>Search Path</u> .....	7-17
<u>Proxy URL</u> .....	7-17

# Table of Contents

## Chapter 7 - GUI Reference

<u>Filename</u> .....	7-18
<u>Dialog Buttons</u> .....	7-19

## Chapter 8 - Command-Line Reference.....8-1

<u>Basic Usage</u> .....	8-1
<u>Options</u> .....	8-1
<u>-d directory</u> .....	8-1
<u>-f filename</u> .....	8-2
<u>-t format</u> .....	8-2
<u>-v</u> .....	8-2
<u>--batch filename.book</u> .....	8-2
<u>--bodycolor color</u> .....	8-2
<u>--bodyfont typeface</u> .....	8-3
<u>--bodyimage filename</u> .....	8-3
<u>--book</u> .....	8-3
<u>--bottom margin</u> .....	8-3
<u>--browserwidth pixels</u> .....	8-3
<u>--charset charset</u> .....	8-4
<u>--color</u> .....	8-4
<u>--compression[=level]</u> .....	8-4
<u>--continuous</u> .....	8-5
<u>--cookies 'name=\"value with space\"; name=value'</u> .....	8-5
<u>--datadir directory</u> .....	8-5
<u>--duplex</u> .....	8-5
<u>--effectduration seconds</u> .....	8-5
<u>--embedfonts</u> .....	8-5
<u>--encryption</u> .....	8-5
<u>--firstpage page</u> .....	8-6
<u>--fontsize size</u> .....	8-6
<u>--fontspacing spacing</u> .....	8-6
<u>--footer lcr</u> .....	8-7
<u>--format format</u> .....	8-8
<u>--gray</u> .....	8-8
<u>--header lcr</u> .....	8-8
<u>--headfootfont font</u> .....	8-9
<u>--headfootsize size</u> .....	8-9
<u>--headingfont typeface</u> .....	8-10
<u>--help</u> .....	8-10
<u>--helpdir directory</u> .....	8-10
<u>--jpeg[=quality]</u> .....	8-10
<u>--landscape</u> .....	8-10
<u>--left margin</u> .....	8-10
<u>--linkcolor color</u> .....	8-10
<u>--links</u> .....	8-11
<u>--linkstyle style</u> .....	8-11
<u>--logoimage filename</u> .....	8-11
<u>--no-compression</u> .....	

# Table of Contents

## **Chapter 8 - Command-Line Reference**

<u>--no-localfiles</u>	8-12
<u>--no-numbered</u>	8-12
<u>--no-pscommands</u>	8-12
<u>--no-strict</u>	8-12
<u>--no-title</u>	8-12
<u>--no-toc</u>	8-12
<u>--no-xrxcomments</u>	8-12
<u>--numbered</u>	8-12
<u>--nup pages</u>	8-12
<u>--outdir directory</u>	8-12
<u>--outfile filename</u>	8-13
<u>--owner-password password</u>	8-13
<u>--pageduration seconds</u>	8-13
<u>--pageeffect effect</u>	8-14
<u>--pagelayout layout</u>	8-15
<u>--pagemode mode</u>	8-15
<u>--path dir1:dir2:dir3:...:dirN</u>	8-15
<u>--permissions permission[,permission,...]</u>	8-16
<u>--portrait</u>	8-16
<u>--pscommands</u>	8-16
<u>--quiet</u>	8-16
<u>--referer url</u>	8-16
<u>--right margin</u>	8-17
<u>--size size</u>	8-17
<u>--strict</u>	8-17
<u>--textcolor color</u>	8-17
<u>--textfont typeface</u>	8-18
<u>--title</u>	8-18
<u>--titlefile filename</u>	8-18
<u>--titleimage filename</u>	8-18
<u>--tocfooter lcr</u>	8-18
<u>--toheader lcr</u>	8-18
<u>--toclevels levels</u>	8-18
<u>--toctitle string</u>	8-19
<u>--top margin</u>	8-19
<u>--user-password password</u>	8-19
<u>--verbose</u>	8-19
<u>--version</u>	8-19
<u>--webpage</u>	8-19
<u>--xrxcomments</u>	8-19
<u>Environment Variables</u>	8-20
<u>HTMLDOC DATA</u>	8-20
<u>HTMLDOC DEBUG</u>	8-20
<u>HTMLDOC HELP</u>	8-20
<u>HTMLDOC NOCGI</u>	8-20
<u>Messages</u>	8-21
<u>BYTES: Message</u>	8-21
<u>DEBUG: Messages</u>	8-21
<u>ERRnnn: Messages</u>	8-21
<u>INFO: Messages</u>	8-21
<u>PAGES: Message</u>	8-21
<u>REMOTEBYTES: Message</u>	8-21

# Table of Contents

## **Chapter 8 - Command-Line Reference**

<u>TIMING: Message</u> .....	8-22
------------------------------	------

## **Appendix A - License Agreement**.....**A-1**

<u>Introduction</u> .....	A-1
<u>Source Code and the GNU GPL</u> .....	A-1
<u>Trademarks</u> .....	A-2
<u>Binary Distribution Rights</u> .....	A-2
<u>Binaries and Support</u> .....	A-2
<u>End-User License Agreement</u> .....	A-3
<u>TERMS AND CONDITIONS OF SOFTWARE LICENSE</u> .....	A-3
<u>LIMITED WARRANTY AND DISCLAIMER OF WARRANTY: LIMITATION OF LIABILITY</u> .....	A-4

## **Appendix B - Book File Format**.....**B-1**

<u>Introduction</u> .....	B-1
<u>The Header</u> .....	B-1
<u>The Options</u> .....	B-2
<u>The Files</u> .....	B-2
<u>Putting It All Together</u> .....	B-2
<u>Older Book Files</u> .....	B-3

## **Appendix C - Release Notes**.....**C-1**

<u>Changes in HTMLDOC v1.8.27</u> .....	C-1
<u>Changes in HTMLDOC v1.8.26</u> .....	C-1
<u>Changes in HTMLDOC v1.8.25</u> .....	C-1
<u>Changes in HTMLDOC v1.8.24</u> .....	C-2
<u>Changes in HTMLDOC v1.8.23</u> .....	C-2
<u>Changes in HTMLDOC v1.8.22</u> .....	C-2
<u>Changes in HTMLDOC v1.8.21</u> .....	C-2
<u>Changes in HTMLDOC v1.8.20</u> .....	C-2
<u>Changes in HTMLDOC v1.8.19</u> .....	C-3
<u>Changes in HTMLDOC v1.8.18</u> .....	C-3
<u>Changes in HTMLDOC v1.8.17</u> .....	C-3
<u>Changes in HTMLDOC v1.8.16</u> .....	C-4
<u>Changes in HTMLDOC v1.8.15</u> .....	C-4
<u>Changes in HTMLDOC v1.8.14</u> .....	C-4
<u>Changes in HTMLDOC v1.8.13</u> .....	C-4
<u>Changes in HTMLDOC v1.8.12</u> .....	C-4
<u>Changes in HTMLDOC v1.8.8</u> .....	C-5

## Table of Contents

### **Appendix D - Compiling HTMLDOC from Source**

<u>Installing with Visual C++</u> .....	C-6
---	-----



# Introduction

This document describes how to use the HTMLDOC software, version 1.8.27. HTMLDOC converts Hyper-Text Markup Language ("HTML") input files into indexed HTML, Adobe® PostScript®, or Adobe Portable Document Format ("PDF") files.

HTMLDOC supports most HTML 3.2 elements, some HTML 4.0 elements, and can generate title and table of contents pages. It does not currently support stylesheets.

HTMLDOC can be used as a standalone application, in a batch document processing environment, or as a web-based report generation application.

No restrictions are placed upon the output produced by HTMLDOC.

HTMLDOC is available both as open source software under the terms of the GNU General Public License and as commercial software under the terms of a traditional commercial End-User License Agreement.

## History

Like many programs HTMLDOC was developed in response to a need our company had for generating high-quality documentation in printed and electronic forms. For a while we used FrameMaker® and a package from *sgi* that generated "compiled" Standard Generalized Markup Language ("SGML") files that could be used by the Electronic Book Technologies ("EBT") documentation products; EBT was bought by INSO who was bought by Stellent™ who apparently has dropped the whole product line. When *sgi* stopped supporting these tools we turned to INSO, but the cost of their tools is prohibitive to small businesses.

In the end we decided to write our own program to generate our documentation. HTML seemed to be the source format of choice since WYSIWYG HTML editors are widely (and freely) available and at worst you

can use a plain text editor. We needed HTML output for documentation on our web server, PDF for customers to read and/or print from their computers, and PostScript for our own printing needs.

The result of our efforts is the HTMLDOC software which is available for Linux®/UNIX®, MacOS® X, and Microsoft® Windows®. Among other things, this software users manual is produced using HTMLDOC.

## Organization of This Manual

This manual is organized into tutorial and reference chapters and appendices:

- [Chapter 1](#) - Installing HTMLDOC
- [Chapter 2](#) - Getting Started
- [Chapter 3](#) - Generating Books
- [Chapter 4](#) - HTMLDOC from the Command-Line
- [Chapter 5](#) - HTMLDOC from a Web Server
- [Chapter 6](#) - HTML Reference
- [Chapter 7](#) - GUI Reference
- [Chapter 8](#) - Command-Line Reference
- [Appendix A](#) - License Agreement
- [Appendix B](#) - Book File Format
- [Appendix C](#) - Release Notes
- [Appendix D](#) - Compiling HTMLDOC from Source

## Support

Commercial support is available from Easy Software Products when you purchase the HTMLDOC Professional Membership. More information is available at the HTMLDOC web page at the following URL:

Solaris is a registered trademark of Sun Microsystems, Inc.

SPARC is a registered trademark of SPARC International, Inc.

UNIX is a registered trademark of the X/Open Company, Ltd.

HTMLDOC is the trademark property of Easy Software Products.

HTMLDOC is copyright 1997-2006 by Easy Software Products. See [Appendix A - License Agreement](#) for the terms of use.

This software is based in part on the work of the Independent JPEG Group and FLTK project.



# Chapter 1 - Installing HTMLDOC

This chapter describes the steps needed to install the commercial version of HTMLDOC on your system. If you are installing HTMLDOC from source code, please see [Appendix D. Compiling HTMLDOC from Source](#).

## Requirements

HTMLDOC requires approximately 4MB of disk space and one of the following environments:

- Microsoft Windows® 2000 or higher
- MacOS® X 10.2 or higher
- Linux® 2.4 or higher
- Solaris® 7 or higher

HTMLDOC may run on other platforms, however we do not provide packages for platforms other than those listed.

## Installing HTMLDOC

The following instructions describe how to install the HTMLDOC software on your system.

### Installing HTMLDOC on Microsoft Windows

HTMLDOC is provided as a Microsoft installer file under Windows. Insert the CD or double-click on the *htmldoc* icon in the *Explorer* window to install HTMLDOC under Windows using the Microsoft software installation wizard (Figure 1-1).







Figure 1-4 - The HTMLDOC License Dialog

Enter the license key that was emailed to you or came on the inside of the HTMLDOC CD-ROM case and click on the *OK* button. Click on the *Close* button to start using the software.

## Uninstalling HTMLDOC

The following instructions describe how to remove the HTMLDOC software from your system.

### Uninstalling HTMLDOC on Microsoft Windows

Open the Control Panel window and double-click on the *Add/Remove Software* icon. When the available software list is displayed, select HTMLDOC and click on the *Remove* button.

# Chapter 2 - Getting Started

This chapter describes how to start HTMLDOC and convert HTML files into PostScript and PDF files.

**Note:**

HTMLDOC currently does not support HTML 4.0 features such as stylesheets or the STYLE, TBODY, THEAD, or TFOOT elements. For more information, please consult [Chapter 6 - HTML Reference](#).

## Starting HTMLDOC

For Windows click:

Start Menu->All Programs->HTMLDOC->HTMLDOC

For MacOS X click:

Applications Folder->HTMLDOC

For Linux click:

Applications Menu->Office->HTMLDOC

or type:

`htmldoc ENTER`

For Solaris click:

Applications Window->ESP->HTMLDOC

or type:

`htmldoc ENTER`

## Choosing a HTML File

The HTMLDOC window (Figure 2-1) shows the list of input files that will be converted. Start by clicking on the *Web Page* radio button (1) to specify that you will be converting a HTML web page file.



Figure





# Chapter 3 - Generating Books

This chapter describes how to create a book using HTML files.

## Overview

While HTMLDOC can convert web pages into PostScript and PDF files, its real strength is generating indexed HTML, PostScript, or PDF books.

HTMLDOC uses HTML heading elements to delineate chapters and headings in a book. The H1 element is used for chapters:

```
<HTML>
<HEAD>
  <TITLE>The Little Computer that Could</TITLE>
</HEAD>
<BODY>
<H1>Chapter 1 - The Little Computer is Born</H1>
...
<H1>Chapter 2 - Little Computer's First Task</H1>
...
</BODY>
</HTML>
```

Sub-headings are marked using the H2 through H6 elements.







# Chapter 4 - HTMLDOC from the Command-Line

This chapter describes how to use HTMLDOC from the command-line to convert web pages and generate books.

## Getting to the Command-Line on Windows

Do the following steps to access the command-line on Windows:

1. Click on *Start* at the bottom left corner of your screen
2. Click on *All Programs*
3. Click on *Accessories*
4. Click on *Command Prompt*

After you have clicked command prompt, your screen should look something like Figure 4-1.



Figure 4-1: Command prompt window

To see what's in this directory, type the following command:

```
dir ENTER
```

You now have a list of available files and directories that you can use. To access a different directory simply type **cd** and the name of the new directory. For example, type the following if you want to access a directory called *Steve*:

```
cd Steve ENTER
```

## The Basics of Command-Line Access

To convert a single web page type:

```
htmldoc --webpage -f output.pdf filename.html ENTER
```

### What Are All These Commands?

## Converting Multiple HTML Files

To convert more than one web page with page breaks between each HTML file, type:

```
htmldoc --webpage -f output.pdf file1.html file2.html ENTER
```

All we are doing is adding another file. In this example we are converting two files: *file1.html* and *file2.html*.

Try this example: Convert *one.html* and *two.html* into a PDF file named *12pdf.pdf*. Again, the answer is on the next line.

Your line command should look like this:

```
htmldoc --webpage -f 12pdf.pdf one.html two.html ENTER
```

We've been using HTML files, but you can also use URLs. For example:

```
htmldoc --webpage -f output.pdf http://slashdot.org/ ENTER
```

## Generating Books

Type one of the following commands to generate a book from one or more HTML files:</

## Setting the Title File

The `--titlefile` option sets the HTML file or image to use on the title page:

```
htmldoc --titlefile filename.bmp ... ENTER
htmldoc --titlefile filename.gif ... ENTER
htmldoc --titlefile filename.jpg ... ENTER
htmldoc --titlefile filename.png ... ENTER
htmldoc --titlefile filename.html ... ENTER
```

HTMLDOC supports BMP, GIF, JPEG, and PNG images, as well as generic HTML text you supply for the title page(s).

## Putting It All Together

```
htmldoc --book -f 12book.pdf 1book.html 2book.html --titlefile bookcover.jpg ENTER
```

Take a look at the entire command line. Dissect the information. Can you see what the new filename is? What are the names of the files being converted? Do you see the titlepage file? What kind of file is your titlefile?

Figure it out? The new file is *12book.pdf*. The files converted were *1book.html* and *2book.html*. A title page was created using the JPEG image file *bookcover.jpg*.

[Chapter 8 - Command Line Reference](#) digs deeper into what you can do with the the command line prompt.

# Chapter 5 - Using HTMLDOC on a Web Server

This chapter describes how to interface HTMLDOC to your web server using CGI and your own server-side scripts and programs.

## The Basics

HTMLDOC can be used in a variety of ways to generate formatted reports on a web server. The most common way is to use HTMLDOC as a CGI program with your web server to provide PDF-formatted output of a web page. Examples are provided for Microsoft IIS and the Apache web servers.

HTMLDOC can also be called from your own server-side scripts and programs. Examples are provided for PHP and Java.

### **WARNING:**

Passing information directly from the web browser to HTMLDOC can potentially expose your system to security risks. Always be sure to "sanitize" any input from the web browser so that filenames, URLs, and options passed to HTMLDOC are not acted on by the shell program or other processes.

## Using HTMLDOC as a CGI Program

HTMLDOC 1.8.24 and higher supports operation as a CGI program. You can copy or symlink the *htmldoc* (all but Windows) or *htmldoc.exe* (Windows) executable to your web server's *cgi-bin* directory and then use it to produce PDF versions of your web pages.

The CGI converts a page on your local server to PDF and sends it to the client's web browser. For example, to convert a page called *superproducts.html* at the following URL:

```
http://servername/superproducts.html
```

and if you installed HTMLDOC in your server's *cgi-bin* directory, you would direct your clients to the following URL:

```
http://servername/cgi-bin/htmldoc/superproducts.html
```

The boldface portion represents the location of the HTMLDOC executable on the web server. You simply place that path before the page you want to convert.

Form data using the GET method can be passed at the end of the URL, for example:

```
http://servername/cgi-bin/htmldoc/superproducts.html?name=value
```

## Server-Side Preferences

When run as a CGI program, HTMLDOC will try to read a book file to set any preferences for the conversion to PDF. For the *superproducts.html* file described previously, HTMLDOC will look at the following URLs for a book file:

```
http://servername/superproducts.html.book  
http://servername/.book  
http://servername/cgi-bin/.book
```

## Configuring HTMLDOC with Microsoft IIS

The IIS web server is configured to run CGI programs by either modifying the permissions of an existing directory or by creating a new virtual directory that allows for execution of programs. Start by running the *Internet Services Manager* program (Figure 5-1):

1. Click on *Start*
2. Click on *Settings*
3. Click on *Control Panel*
4. Double-click on *Administrative Tools*
5. Double-click on *Internet Services Manager*

After the *Internet Services Manager* window (Figure 5-1) appears, perform the following steps to add a virtual folder for HTMLDOC:

1. Click on your server in the list to show the default web site service in the list (Figure 5-2)
2. Choose *New->Virtual Directory* from the *Action* menu (Figure 5-3)
3. Click *Next* when the *Virtual Directory Creation Wizard* window appears (Figure 5-4)
4. Enter the name `html doc` in the *Alias* field and click *Next* (Figure 5-5)
5. Enter the HTMLDOC program folder in the *Directory* field and click *Next* (Figure 5-6)
6. Check the *Execute (such as ISAPI applications or CGI)* box and click *Next*







Figure 5-6: Entering the HTMLDOC Program Folder



Figure 5-7: Enabling CGI Mode



Finally, double-click the *My Computer* icon on the desktop or start the *Windows Explorer*. When the explorer window appears, perform the following steps to provide write access to the Windows temporary folder:

1. Open the windows temporary file folder, typically *C:\WINDOWS\TEMP*
2. Choose *Properties* from the *File* menu
3. Click on the *Security* tab
4. Click *Add...*, enter the username for the web server, typically "SERVER\IUSR\_SERVER" where "SERVER" is the name you gave your server, and click *OK*
5. Click on the username you just added in the list
6. Check the *Read* and *Write* permissions
7. Click *OK* to save the changes

## Using HTMLDOC From Server-Side Scripts and Programs

To make this work the CGI script or program must send the appropriate HTTP attributes, the required empty line to signify the beginning of the document, and then execute the HTMLDOC program to generate the HTML, PostScript, or PDF file as needed. Since HTMLDOC looks for CGI environment variables when it is run, you must also set the HTMLDOC\_NOCGI environment variable to a value of 1 before running HTMLDOC from your CGI script or program.

Another way to generate PDF files from your reports is to use HTMLDOC as a "portal" application. When used as a portal, HTMLDOC automatically retrieves the named document or report from your server and passes a PDF version to the web browser. See the next sections for more information.

## Calling HTMLDOC from a Shell Script

Shell scripts are probably the easiest to work with, but are normally limited to GET type requests. Here is a script called *topdf* that acts as a portal, converting the named file to PDF:

```
#!/bin/sh
#
# Sample "portal" script to convert the named HTML file to PDF on-the-fly.
#
# Usage: http://www.domain.com/path/topdf/path/filename.html
#
#
# Tell HTMLDOC not to run in CGI mode...
#

HTMLDOC_NOCGI=1; export HTMLDOC_NOCGI

#
# The "options" variable contains any options you want to pass to HTMLDOC.
#

options='-t pdf --webpage --header ... --footer ...'

#
# Tell the browser to expect a PDF file...
#

echo "Content-Type: application/pdf"
echo ""

#
# Run HTMLDOC to generate the PDF file...
#

htmldoc $options http://${SERVER_NAME}:${SERVER_PORT}$PATH_INFO
```

Users of this CGI would reference the URL "http://www.domain.com/topdf.cgi/index.html" to generate a PDF file of the site's home page.

The *options* variable in the script can be set to use any supported command-line option for HTMLDOC; for a complete list see [Chapter 8 - Command-Line Reference](#).

## Calling HTMLDOC from Perl

Perl scripts offer the ability to generate more complex reports, pull data from databases, etc. The easiest way to interface Perl scripts with HTMLDOC is to write a report to a temporary file and then execute HTMLDOC to generate the PDF file.

Here is a simple Perl subroutine that can be used to write a PDF report to the HTTP client:

```
sub topdf {
    # Get the filename argument...
    my $filename = shift;

    # Make stdout unbuffered...
    select(STDOUT); $| = 1;

    # Tell HTMLDOC not to run in CGI mode...
    $ENV{HTMLDOC_NOCGI} = 1;

    # Write the content type to the client...
    print "Content-Type: application/pdf\n\n";

    # Run HTMLDOC to provide the PDF file to the user...
    system "htmldoc -t pdf --quiet --webpage $filename";
}
```

## Calling HTMLDOC from PHP

PHP is quickly becoming the most popular server-side scripting language available. PHP provides a `passthru()` function that can be used to run HTMLDOC. This combined with the `header()` function can be used to provide on-the-fly reports in PDF format.

Here is a simple PHP function that can be used to convert a HTML report to PDF and send it to the HTTP client:

```
function topdf($filename, $options = "") {
    # Tell HTMLDOC not to run in CGI mode...
    putenv
```

```

function bad_url($url) {
    // See if the URL starts with http: or https:...
    if (strncmp($url, "http://", 7) != 0 &&
        strncmp($url, "https://", 8) != 0) {
        return 1;
    }

    // Check for bad characters in the URL...
    $len = strlen($url);
    for ($i = 0; $i < $len; $i++) {
        if (!strchr("~_*/:~*()/:%?+-&@;=, $.", $url[$i]) &&
            !ctype_alnum($url[$i])) {
            return 1;
        }
    }

    return 0;
}

```

Another method is to use the `escapeshellarg()` function provided with PHP 4.0.3 and higher to generate a quoted shell argument for HTMLDOC.

To make a "portal" script, add the following code to complete the example:

```

global $SERVER_NAME;
global $SERVER_PORT;
global $PATH_INFO;
global $QUERY_STRING;

if ($QUERY_STRING != "") {
    $url = "http://${SERVER_NAME}:${SERVER_PORT}${PATH_INFO}?${QUERY_STRING}";
} else {
    $url = "http://${SERVER_NAME}:${SERVER_PORT}${PATH_INFO}";
}

if (bad_url($url)) {
    print("<html><head><title>Bad URL</title></head>\n"
        . "<body><h1>Bad URL</h1>\n"
        . "<p>The URL <b><tt>$url</tt></b> is bad.</p>\n"
        . "</body></html>\n");
} else {
    topdf($url);
}

```

## Calling HTMLDOC from C

C programs offer the best flexibility and easily supports on-the-fly report generation without the need for temporary files.

Here are some simple C functions that can be used to generate a PDF report to the HTTP client from a temporary file or pipe:

```
#include <stdio.h>
#include <stdlib.h>

/* topdf() - convert a HTML file to PDF */
FILE *topdf(const char *filename)          /* I - HTML file to convert */
{
    char      command[1024];               /* Command to execute */

    /*
     * Tell HTMLDOC not to run in CGI mode...
     */

    putenv("HTMLDOC_NOCGI=1");

    /*
     * Write the content type to the client...
     */

    puts("Content-Type: application/pdf\n");

    /*
     * Run HTMLDOC to provide the PDF file to the user...
     */

    sprintf(command, "htmldoc --quiet -t pdf --webpage %s", filename);

    return (popen(command, "w"));
}

/* topdf2() - pipe HTML output to HTMLDOC for conversion to PDF */
FILE *topdf2(void)
{
    /*
     * Tell HTMLDOC not to run in CGI mode...
     */

    putenv("HTMLDOC_NOCGI=1");

    /*
     * Write the content type to the client...
     */

    puts("Content-Type: application/pdf\n");

    /*
     * Open a pipe to HTMLDOC...
     */

    return (popen("htmldoc --quiet -t pdf --webpage -", "w"));
}
```

## Calling HTMLDOC from Java

Java programs are a portable way to add PDF support to your web server. Here is a class called *htmldoc* that acts as a portal, converting the named file to PDF. It can also be called by your Java servlets to process an HTML file and send the result to the client in PDF format:

```
class htmldoc
{
    // Convert named file to PDF on stdout...
    public static int topdf(String filename)// I - Name of file to convert
    {
        String          command;          // Command string
        Process          process;          // Process for HTMLDOC
        Runtime          runtime;          // Local runtime object
        java.io.InputStream input;         // Output from HTMLDOC
        byte             buffer [];        // Buffer for output data
        int              bytes;            // Number of bytes

        // First tell the client that we will be sending PDF...
        System.out.print("Content-type: application/pdf\n\n");

        // Construct the command string
        command = "htmldoc --quiet --jpeg --webpage -t pdf --left 36 " +
            "--header .t. --footer .l. " + filename;

        // Run the process and wait for it to complete...
        runtime = Runtime.getRuntime();

        try
        {
            // Create a new HTMLDOC process...
            process = runtime.exec(command);

            // Get stdout from the process and a buffer for the data...
            input = process.getInputStream();
            buffer = new byte[8192];

            // Read output from HTMLDOC until we have it all...
            while ((bytes = input.read(buffer)) > 0)
                System.out.write(buffer, 0, bytes);

            // Return the exit status from HTMLDOC...
            return (process.waitFor
```

```
(server_port = System.getProperty("SERVER_PORT")) != null &&
(path_info = System.getProperty("PATH_INFO")) != null)
{
    // Construct a URL for the resource specified...
    filename = "http://" + server_name + ":" + server_port + path_info;

    if ((query_string = System.getProperty("QUERY_STRING")) != null)
    {
        filename = filename + "?" + query_string;
    }
}
else if (args.length == 1)
{
    // Pull the filename from the command-line...
    filename = args[0];
}
else
{
    // Error - no args or env variables!
    System.err.print("Usage: htmldoc.class filename\n");
    return;
}

// Convert the file to PDF and send to the web client...
topdf(filename);
}
```

# Chapter 6 - HTML Reference

This chapter defines all of the HTML elements and attributes that are recognized and supported by HTMLDOC.

## General Usage

There are two types of HTML files - structured documents using headings (H1, H2, etc.) which HTMLDOC calls "books", and unstructured documents that do not use headings which HTMLDOC calls "web pages".

A very common mistake is to try converting a web page using:

```
htmldoc -f filename.pdf filename.html
```

which will likely produce a PDF file with no pages. To convert web page files you **must** use the `--webpage` option at the command-line or choose *Web Page* in the input tab of the GUI.

**Note:**

HTMLDOC does not support HTML 4.0 elements, attributes, stylesheets, or scripting.

## Elements

The following HTML elements are recognized by HTMLDOC:

Element	Version	Supported?	Notes
!DOCTYPE	3.0	Yes	DTD is ignored
A	1.0	Yes	<a href="#">See Below</a>
ACRONYM	2.0	Yes	No font change
ADDRESS	2.0	Yes	
AREA	2.0	No	
B	1.0	Yes	
BASE	2.0	No	
BASEFONT	1.0	No	
BIG	2.0	Yes	
BLINK	2.0	No	
BLOCKQUOTE	2.0	Yes	
BODY	1.0	Yes	
BR	2.0	Yes	
CAPTION	2.0	Yes	
CENTER	2.0	Yes	
CITE	2.0	Yes	Italic/Oblique
CODE	2.0	Yes	Courier
DD	2.0	Yes	
DEL	2.0	Yes	Strikethrough
DFN	2.0	Yes	Helvetica
DIR	2.0	Yes	
DIV	3.2	Yes	
DL	2.0	Yes	
DT	2.0	Yes	Italic/Oblique
EM	2.0	Yes	Italic/Oblique
EMBED	2.0	Yes	HTML Only
FONT	2.0	Yes	<a href="#">See Below</a>
FORM	2.0	No	
FRAME	3.2	No	

Element	Version	Supported?	Notes
FRAMESET	3.2	No	
H1	1.0	Yes	Boldface, <a href="#">See Below</a>
H2	1.0	Yes	Boldface, <a href="#">See Below</a>
H3	1.0	Yes	Boldface, <a href="#">See Below</a>
H4	1.0	Yes	Boldface, <a href="#">See Below</a>
H5	1.0	Yes	Boldface, <a href="#">See Below</a>
H6	1.0	Yes	Boldface, <a href="#">See Below</a>
HEAD	1.0	Yes	
HR	1.0	Yes	<a href="#">See Below</a>
HTML	1.0	Yes	
I	1.0	Yes	
IMG	1.0	Yes	<a href="#">See Below</a>
INPUT	2.0	No	
INS	2.0	Yes	Underline
ISINDEX	2.0	No	
KBD	2.0	Yes	Courier Bold
LI	2.0	Yes	
LINK	2.0	No	
MAP	2.0	No	
MENU	2.0	Yes	
META	2.0	Yes	<a href="#">See Below</a>
MULTICOL	N3.0	No	
NOBR	1.0	No	
NOFRAMES	3.2	No	
OL	2.0	Yes	
OPTION	2.0	No	
P	1.0	Yes	
PRE	1.0	Yes	
S	2.0	Yes	Strikethrough
SAMP	2.0	Yes	Courier
SCRIPT	2.0	No	

Element	Version	Supported?	Notes
SELECT	2.0	No	
SMALL	2.0	Yes	
SPACER	N3.0	Yes	
STRIKE	2.0	Yes	
STRONG	2.0	Yes	Boldface Italic/Oblique
SUB	2.0	Yes	Reduced Fontsize
SUP	2.0	Yes	Reduced Fontsize
TABLE	2.0	Yes	<a href="#">See Below</a>
TD	2.0	Yes	
TEXTAREA	2.0	No	
TH	2.0	Yes	Boldface Center
TITLE	2.0	Yes	
TR	2.0	Yes	
TT	2.0	Yes	Courier
U	1.0	Yes	
UL	2.0	Yes	
VAR	2.0	Yes	Helvetica Oblique
WBR	1.0	No	



## Header/Footer Strings

The HEADER and FOOTER comments allow you to set an arbitrary string of text for the left, center, and right headers and footers. Each string consists of plain text; special values or strings can be inserted using the dollar sign (\$):

**\$\$**  
 Inserts a single dollar sign in the header.

**\$CHAPTER**  
 Inserts the current chapter heading.

**\$CHAPTERPAGE**  
**\$CHAPTERPAGE ( format )**  
 Inserts the current page number within a chapter or file. When a format is specified, uses that numeric format (1 = decimal, i = lowercase roman numerals, I = uppercase roman numerals, a = lowercase ascii, A = uppercase ascii) for the page numbers.

**\$CHAPTERPAGES**  
**\$CHAPTERPAGES ( format )**  
 Inserts the total page count within a chapter or file. When a format is specified, uses that numeric format (1 = decimal, i = lowercase roman numerals, I = uppercase roman numerals, a = lowercase ascii, A = uppercase ascii) for the page count.

**\$DATE**  
 Inserts the current date.

**\$HEADING**  
 Inserts the current heading.

**\$LOGOIMAGE**  
 Inserts the logo image; all other text in the string will be ignored.

**\$PAGE**  
**\$PAGE ( format )**  
 Inserts the current page number. When a format is specified, uses that numeric format (1 = decimal, i = lowercase roman numerals, I = uppercase roman numerals, a = lowercase ascii, A = uppercase ascii) for the page numbers.

**\$PAGES**  
**\$PAGES ( format )**  
 Inserts the total page count. When a format is specified, uses that numeric format (1 = decimal, i = lowercase roman numerals, I = uppercase roman numerals, a = lowercase ascii, A = uppercase ascii) for the page count.

**\$TIME**  
 Inserts

## FONT Attributes

Limited typeface specification is currently supported to ensure portability across platforms and for older PostScript printers:

Requested Font	Actual Font
Arial	Helvetica
Courier	Courier
Dingbats	Dingbats
Helvetica	Helvetica
Monospace	DejaVu Sans Mono
Sans	DejaVu Sans
Serif	DejaVu Serif
Symbol	Symbol
Times	Times

All other unrecognized typefaces are silently ignored.

## Headings

Currently HTMLDOC supports a maximum of 1000 chapters (H1 headings). This limit can be increased by changing the MAX\_CHAPTERS constant in the *config.h* file included with the source code.

All chapters start with a top-level heading (H1) markup. Any headings within a chapter must be of a lower level (H2 to H15). Each chapter starts a new page or the next odd-numbered page if duplexing is selected.

**Note:**

```
...
<H4>Sub-Sub-Section Heading 3</H4>
...
```

the table-of-contents that is generated will show:

### Chapter Heading

- ◆ Section Heading 1
- ◆ Section Heading 2
  - ◇ Sub-Section Heading 1
    - Sub-Sub-Section Heading 1
    - Sub-Sub-Section Heading 2
  - ◇ Sub-Section Heading 2
    - Sub-Sub-Section Heading 3
- ◆ Section Heading 3

## Numbered Headings

When the numbered headings option is enabled, HTMLDOC recognizes the following additional attributes for all heading elements:

VALUE=" # "

Specifies the starting value for this heading level (default is "1" for all new levels).

TYPE=" 1 "

## META Attributes

HTMLDOC supports the following META attributes for the title page and document information:

```
<META NAME="AUTHOR" CONTENT=" . . . "
    Specifies the document author.
<META NAME="COPYRIGHT" CONTENT=" . . . "
    Specifies the document copyright.
<META NAME="DOCNUMBER" CONTENT=" . . . "
    Specifies the document number.
<META NAME="GENERATOR" CONTENT=" . . . "
    Specifies the application that generated the HTML file.
<META NAME="KEYWORDS" CONTENT=" . . . "
    Specifies document search keywords.
<META NAME="SUBJECT" CONTENT=" . . . "
    Specifies document subject.
```

## Page Breaks

HTMLDOC supports four new page comments to specify page breaks. In addition, the older BREAK attribute is still supported by the HR element:

```
<HR BREAK>
```

Support for the BREAK attribute is deprecated and will be removed in a future release of HTMLDOC.

## Tables

Currently HTMLDOC supports a maximum of 200 columns within a single table. This limit can be increased by changing the MAX\_COLUMNS constant in the *config.h* file included with the source



# Chapter 7 - GUI Reference

This chapter describes all of the GUI controls in HTMLDOC.

## The HTMLDOC GUI

The HTMLDOC GUI (Figures 7-1 through 7-11) is contained in a single window showing the input, output, and generation options. At the bottom are buttons to load, save, and generate documents.

### Document File Operations

HTMLDOC stores the HTML files, settings, and options in .BOOK files. The buttons on the bottom of the HTMLDOC window allow you to manage these files and generate formatted documents.

#### New

The *New* button starts a new document. A confirmation dialog will appear if you have not saved the changes to the existing document.

#### Open...

The *Open...* button retrieves a document that you have saved previously. A file chooser dialog is displayed that allows you to pick an existing book file.

## Save

The **Save** button saves the current document. A file chooser dialog is displayed if there is no filename assigned to the current document.

**Note:** Saving a document is not the same as *generating* a document. The book files saved to disk by the **Save** and **Save As...** buttons are **not** the final HTML, PDF, or PostScript output files. You generate those files by clicking on the *Generate* button.

## Save As...

The **Save As...** button saves the current document to a new file. A file chooser dialog is displayed to allow you to specify the new document filename.

**Note:** Saving a document is not the same as *generating* a document. The book files saved to disk by the **Save** and **Save As...** buttons are **not** the final HTML, PDF, or PostScript output files. You generate those files by clicking on the *Generate* button.

## Generate

The *Generate* button generates the current document, creating the specified HTML, PDF, or PostScript file(s) as needed. The progress meter at the bottom of the window will show the progress as each page or file is formatted and written.

**Note:** Generating a document is not the same as **saving** a document. To save the current HTML files and settings in the HTMLDOC GUI, click on the **Save** or <



## Delete Files

The *Delete Files* button removes the selected files from the *Input Files* list. Select one or more files in the *Input Files* list to enable the *Delete Files* button.

The *Delete Files* button only removes the files from the *Input Files* list. The files are **not** removed from disk.

## Move Up

The *Move Up* button moves the selected files in the *Input Files* list up one line in the list. To enable the *Move Up* button select one or more files in the *Input Files* list.

## Move Down

The *Move Down* button moves the selected files in the *Input Files* list down one line in the list. To enable the *Move Down* button select one or more files in the *Input Files* list.

## Logo Image

The *Logo Image* field contains the filename for an image to be shown in the header or footer of pages, and in the navigation bar of HTML files.

Click on the *Browse...* button to select a logo image file using the [file chooser](#) dialog.

## Title File



## Compression

The *Compression* slider controls the amount of compression that is used when writing PDF or Level 3 PostScript output.

**Note:** HTMLDOC uses Flate compression, which is not encumbered by patents and is also used by the popular PKZIP and gzip programs. Flate is a lossless compression algorithm (that is, you get back exactly what you put in) that performs very well on indexed images and text.

## JPEG Quality

The *JPEG Quality* slider controls the quality level used when writing continuous-tone images with JPEG compression.



## Header and Footer

Select the desired text in each of the option buttons to customize the header and footer for the document/body pages. The left-most option buttons set the text that is left-justified, while the middle buttons set the text that is centered and the right buttons set the text that is right-justified. Each choice corresponds to the following text:

Choice	Description
Blank	The field should be blank.
Title	The field should contain the document title.
Chapter Title	The field should contain the current chapter title.
Heading	The field should contain the current heading.
Logo	The field should contain the logo image.
1,2,3,...	The field should contain the current page number in decimal format (1, 2, 3, ...)
i,ii,iii,...	The field should contain the current page number in lowercase roman numerals (i, ii, iii, ...)
I,II,III,...	The field should contain the current page number in uppercase roman numerals (I, II, III, ...)
a,b,c,...	The field should contain the current page number using lowercase letters.
A,B,C,...	The field should contain the current page number using UPPERCASE letters.
Chapter Page	The field should contain the current chapter page number.
1/N,2/N,...	The field should contain the current and total number of pages (n/N).
1/C,2/C,...	The field should contain the current and total number of pages in the chapter (n/N).
Date	The field should contain the current date (formatted for the current locale).
Time	The field should contain the current time (formatted for the current locale).











## Page Mode

The *Page Mode* option button controls the initial viewing mode for the document. Click on the option button to set the page mode.

The *Document* page mode displays only the document pages. The *Outline* page mode displays the table-of-contents outline as well as the document pages. The *Full-Screen* page mode displays the document pages on the whole screen; this mode is used primarily for presentations.

## Page Layout

The *Page Layout* option button controls the initial layout of document pages on the screen. Click on the option button to set the page layout.

The *Single* page layout displays a single page at a time. The *One Column* page layout displays a single column of pages at a time. The *Two Column Left* and *Two Column Right* page layouts display two columns of pages at a time; the first page is displayed in the left or right column as selected.

## First Page

The *First Page* option button controls the initial page that is displayed. Click on the option button to choose the first page.

## Page Effect

The *Page Effect* option button controls the page effect that is displayed in *Full-Screen* mode. Click on the option button to select a page effect.





The default browser width is 680 pixels which corresponds roughly to a 96 DPI display. The browser width is only used when generating PostScript or PDF files.

## Search Path

The *Search Path* field specifies a search path for files that are loaded by HTMLDOC. It is usually used to get images that use absolute server paths to load.

Directories are separated by the semicolon (;) so that drive letters (and eventually URLs) can be specified.

## Proxy URL

The *Proxy URL* field specifies a URL for a HTTP proxy server.

## Tooltips

The *Tooltips* check button controls the appearance of tooltip windows over GUI controls.

## Modern Look

The *Modern Look* check button controls the appearance of the GUI controls.

## Strict HTML

The *Strict HTML* check button controls strict HTML conformance checking. When checked, HTML elements that are improperly nested and dangling close elements will produce error messages.



The button bar along the top of the filename allows you to view each directory in the filename. Click on any of the segments to display the corresponding directory.

## **Dialog Buttons**

The dialog buttons (5) close the file chooser dialog window. Click on the *OK* button to accept your selections or the *Cancel* button to reject your selections and cancel the file operation.



# Chapter 8 - Command-Line Reference

This chapter describes all of the command-line options supported by HTMLDOC.

## Basic Usage

The basic command-line usage for HTMLDOC is:

```
% htmldoc options filename1.html ... filenameN.html ENTER  
% htmldoc options filename.book ENTER
```

The first form converts the named HTML files to the specified output format immediately. The second form loads the specified `.book` file and displays the HTMLDOC window, allowing a user to make changes and/or generate the document interactively.

If no output file or directory is specified, then all output is sent to the standard output file.

On return, HTMLDOC returns an exit code of 0 if it was successful and non-zero if there were errors.

## Options

The following command-line options are recognized by HTMLDOC.

### **-d directory**

The `-d` option specifies an output directory for the document files.

This option is not compatible with the PDF output format.

**-f filename**

The `-f` option specifies an output file for the document.

**-t format**

The `-t` option specifies the output format for the document and can be one of the following:

Format	Description
html	Generate one or more indexed HTML files.
htmlsep	Generate separate HTML files for each heading in the table-of-contents.
pdf	Generate a PDF file (default version - 1.3).
pdf11	Generate a PDF 1.1 file for Acrobat Reader 2.0.
pdf12	Generate a PDF 1.2 file for Acrobat Reader 3.0.
pdf13	Generate a PDF 1.3 file for Acrobat Reader 4.0.
pdf14	Generate a PDF 1.4 file for Acrobat Reader 5.0.
ps	Generate one or more PostScript files (default level - 2).
ps1	Generate one or more Level 1 PostScript files.
ps2	Generate one or more Level 2 PostScript files.
ps3	Generate one or more Level 3 PostScript files.

**-v**

The

**--bodyfont typeface**

The `--bodyfont` option specifies the default text font used for text in the document body. The `typeface` parameter can be one of the following:

<b>typeface</b>	<b>Actual Font</b>
Arial	Helvetica
Courier	Courier
Helvetica	Helvetica
Monospace	DejaVu Sans Mono
Sans	DevaVu Sans
Serif	DejaVu Serif
Times	Times

**--bodyimage filename**

The `--bodyimage` option specifies the background image for all pages in the document. The supported formats are BMP, GIF, JPEG, and PNG.

**--book**

The `--book` option specifies that the input files comprise a book with chapters and headings.

**--bottom margin**

The `--bottom` option specifies the bottom margin. The default units are points (1 point = 1/7





**--firstpage page**

The `--firstpage` option specifies the first page that will be displayed in a PDF file. The `page` parameter can be one of the following:

<b>page</b>	<b>Description</b>
p1	The first page of the document.
toc	The first page of the table-of-contents.
c1	The first page of chapter 1.

This option is only available when generating PDF files.

**--fontsize size**

The `--fontsize` option specifies the base font size for the entire document in points (1 point = 1/72nd inch).

**--fontspacing spacing**

The `--fontspacing` option specifies the line spacing for the entire document as a multiplier of the base font size. A `spacing` value of 1 makes each line of text the same height as the font.



**--format format**

The `--format` option specifies the output format for the document and can be one of the following:

Format	Description
html	Generate one or more indexed HTML files.
htmlsep	Generate separate HTML files for each heading in the table-of-contents.
pdf	Generate a PDF file (default version - 1.3).
pdf11	Generate a PDF 1.1 file for Acrobat Reader 2.0.
pdf12	Generate a PDF 1.2 file for Acrobat Reader 3.0.
pdf13	Generate a PDF 1.3 file for Acrobat Reader 4.0.
pdf14	Generate a PDF 1.4 file for Acrobat Reader 5.0.
ps	Generate one or more PostScript files (default level - 2).
ps1	Generate one or more Level 1 PostScript files.
ps2	Generate one or more Level 2 PostScript files.
ps3	Generate one or more Level 3 PostScript files.

**--gray**

The `--gray` option specifies that grayscale output is desired.

This option is only available when generating PostScript or PDF files.

**--headfootfont font**

The `--headfootfont` option specifies the font that is used for the header and footer text. The `font` parameter can be one of the following:

- Courier
- Courier-Bold
- Courier-Oblique
- Courier-BoldOblique
- Helvetica
- Helvetica-Bold
- Helvetica-Oblique
- Helvetica-BoldOblique
- Monospace
- Monospace-Bold
- Monospace-Oblique
- Monospace-BoldOblique
- Sans
- Sans-Bold
- Sans-Oblique
- Sans-BoldOblique
- Serif
- Serif-Roman
- Serif-Bold
- Serif-Italic
- Serif-BoldItalic
- Times
- Times-Roman
- Times-Bold
- Times-Italic
- Times-BoldItalic

This option is only available when generating PostScript or PDF files.

**--headfootsize size**

The `--headfootsize` option sets the size of the header and footer text in points (1 point = 1/72nd inch).

This option is only available when generating PostScript or PDF files.







### **--outfile filename**

The `--outfile` option specifies an output file for the document.

### **--owner-password password**

The `--owner-password` option specifies the owner password for a PDF file. If not specified or the empty string (""), a random password is generated.

This option is only available when generating PDF files.

### **--pageduration seconds**

The `--pageduration` option specifies the number of seconds that each page will be displayed in the document.

This option is only available when generating PDF files.

















## **TIMING: Message**

The `TIMING:` message specifies the load, render, and total time in seconds for the current command. This message is only displayed if the `HTMLDOC_DEBUG` environment variable has the keyword `timing` or `all`.



























































