HTMLDOC Users Manual

Michael R Sweet
Copyright (C) 1997-2020, All Rights Reserved.

Table of Contents

Chapter 1 - Introduction........1-1
History........1-1
Organization of This Manual........1-2
Encryption Support........1-2
Legal Stuff........1-2

Chapter 2 - Using HTMLDOC........2-1
Using the HTMLDOC GUI........2-1
Generating Books........2-1
Using the HTMLDOC Command........2-2
Generating Books........2-3
Setting the Title File........2-3
Using HTMLDOC on a Web Server........2-3
CGI Mode........2-3
Using HTMLDOC From Server-Side Scripts and Programs........2-6

Chapter 3 - Command-Line Reference........3-1
Basic Usage........3-1
Options........3-1
-d directory........3-1
-f filename........3-2
-t format........3-2
-v........3-2
--batch filename.book........3-2
--bodycolor color........3-2
--bodyfont typeface........3-3
--bodyimage filename........3-3
--book........3-3
--bottom margin........3-3
--browserwidth pixels........3-3
--charset charset........3-4
--color........3-4
--compression[=level]........3-5
--continuous........3-5
--cookies 'name=\"value with space\"; name=value'........3-5
--datadir directory........3-5
--duplex........3-5
--effectduration seconds........3-5
--embedfonts........3-5
--encryption........3-5
--firstpage page........3-6
--fontsize size........3-6
--fontspacing spacing........3-6
--footer lcr........3-7
--format format........3-8
--gray........3-8
--header lcr........3-8
--header1 lcr........3-8
--headfootfont font........3-9
--headfootsize size........3-9
--headingfont typeface........3-10
--help........3-10
--helpdir directory........3-10

Table of Contents

Chapter 3 - Command-Line Reference
--hfimageN filename....3-10
--jpeg[=quality]....3-10
--landscape....3-10
--left margin....3-10
--letterhead filename....3-11
--linkcolor color....3-11
--links....3-11
--linkstyle style....3-11
--logoimage filename....3-11
--no-compression....3-11
--no-duplex....3-11
--no-embedfonts....3-11
--no-encryption....3-11
--no-jpeg....3-12
--no-links....3-12
--no-localfiles....3-12
--no-numbered....3-12
--no-pscommands....3-12
--no-strict....3-12
--no-title....3-12
--no-toc....3-12
--no-xrxcomments....3-12
--numbered....3-12
--nup pages....3-12
--outdir directory....3-13
--outfile filename....3-13
--owner-password password....3-13
--pageduration seconds....3-13
--pageeffect effect....3-14
--pagelayout layout....3-15
--pagemode mode....3-15
--path dir1;dir2;dir3;...;dirN....3-15
--permissions permission[,permission,...]....3-16
--portrait....3-16
--pscommands....3-16
--quiet....3-16
--referer url....3-16
--right margin....3-17
--size size....3-17
--strict....3-17
--textcolor color....3-17
--textfont typeface....3-18
--title....3-18
--titlefile filename....3-18
--titleimage filename....3-18
--tocfooter lcr....3-18
--tocheader lcr....3-18
--toclevels levels....3-18
--toctitle string....3-19
--top margin....3-19
--user-password password....3-19
--verbose....3-19

Table of Contents

Chapter 3 - Command-Line Reference
--version....3-19
--webpage....3-19
--xrxcomments....3-19
Environment Variables....3-20
HTMLDOC_DATA....3-20
HTMLDOC_DEBUG....3-20
HTMLDOC_HELP....3-20
HTMLDOC_NOCGI....3-20
Messages....3-21
BYTES: Message....3-21
DEBUG: Messages....3-21
ERRnnn: Messages....3-21
INFO: Messages....3-21
PAGES: Message....3-21
REMOTEBYTES: Message....3-21
TIMING: Message....3-22

Chapter 4 - HTML Reference....4-1
General Usage....4-1
Elements....4-2
Comments....4-4
Header/Footer Strings....4-6
FONT Attributes....4-7
Headings....4-7
Numbered Headings....4-8
Images....4-8
Links....4-8
META Attributes....4-9
Tables....4-9

Chapter 5 - Markdown Reference....5-1
General Syntax....5-1
Metadata Syntax....5-2
Link Targets and @ Links....5-2
Table Syntax....5-3

Appendix A - License Agreement....A-1

Appendix B - Book File Format....B-1
Introduction....B-1
The Header....B-1
The Options....B-2
The Files....B-2
Putting It All Together....B-2

Chapter 1 - Introduction

This document describes how to use the HTMLDOC software. HTMLDOC reads HTML and Markdown source files or web pages and generates corresponding EPUB, HTML, PostScript, or PDF files with an optional table of contents. HTMLDOC can be used as a standalone application, in a batch document processing environment, or as a web-based report generation application.

HTMLDOC is open source software under the terms of version 2 of the GNU General Public License. No restrictions are placed upon the output produced by HTMLDOC.

History

Like many programs, I developed HTMLDOC in response to a need my company had for generating high-quality documentation in printed and electronic forms. For a while I 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 I turned to INSO, but the cost of their tools was prohibitive to my small business.

In the end I decided to write my own program to generate the 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. I needed HTML output for documentation on my web server, PDF for customers to read and/or print from their computers, and PostScript for printing needs.

The result of my efforts is the HTMLDOC software which runs on Linux®, macOS®, Microsoft® Windows®, and most UNIX® operating systems. Among other things, this software users manual is produced using HTMLDOC.

HTMLDOC used to be available under a commercial end-user license agreement from my former company, Easy Software Products. While that company is no longer in business, I continue to maintain HTMLDOC in my spare time.

Organization of This Manual

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

- Chapter 1 - Introduction
- Chapter 2 - Using HTMLDOC
- Chapter 3 - Command-Line Reference
- Chapter 4 - HTML Reference
- Chapter 5 - Markdown Reference
- Appendix A - License Agreement
- Appendix B - Book File Format

Encryption Support

HTMLDOC includes code to encrypt PDF document files using the RC4 algorithm with up to a 128-bit key. While this software and code may be freely used and exported under current US laws, other countries may restrict your use and possession of this code and software.

Legal Stuff

HTMLDOC is copyright © 1997-2020 by Michael R Sweet. 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 2 - Using HTMLDOC

This chapter describes the basics of how to use HTMLDOC to convert HTML and Markdown files into PostScript and PDF files.

> **Note:** HTMLDOC currently does not support HTML 4.0 features such as stylesheets or the `STYLE` element. For more information, please consult Chapter 4 - HTML Reference.

Using the HTMLDOC GUI

After opening the HTMLDOC application, the HTMLDOC window will appear with the *Input* tab selected. Click on the *Web Page* radio button to specify that you will be converting a web page file. Then choose a file for conversion by clicking on the *Add Files...* button.

Now that you've chosen a file to be converted, click on the *Output* tab to set the output file and format. Finally, click on the *Generate* button at the bottom of the HTMLDOC window to convert the HTML file.

Generating Books

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

> **Note:** When using book mode, HTMLDOC starts rendering with the first `H1` element. Any text, images, tables, and other viewable elements that precede the first `H1` element are silently ignored. Because of this, make sure you have an `H1` element in your HTML file, otherwise HTMLDOC will not convert anything.

Start by clicking on the *Book* radio button to specify you'll be converting one or more files into a book. Then add one or more HTML or Markdown files by clicking on the *Add Files...* button.

HTMLDOC will automatically create a title page for you unless you specify a *Title File/Image*. When the title file is HTML or Markdown, the contents are formatted to produce title page(s). When the title file is an image, the image is centered on the title page with automatically generate content based on the title and other metadata.

After providing all of the input files, click on the *Output* tab to select the output format and file. Finally, click on the *Generate* button to generate the book.

Using the HTMLDOC Command

To convert a single web page type:

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

`htmldoc` is the name of the software.

The `--webpage` option specifies unstructured files with page breaks between each file.

The `-f` option specifies the output file name (`output.pdf`). In this example it is a PDF file.

`Filename.html` is the name of the file that you want to be converted.

To convert more than one web page with page breaks between each file, list each of the files on the end:

```
htmldoc --webpage -f output.pdf file1.html file2.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 files:

```
htmldoc --book -f output.html file1.html file2.html ENTER
htmldoc --book -f output.pdf file1.html file2.html ENTER
htmldoc --book -f output.ps file1.html file2.html ENTER
```

The `--book` option specifies that the input files are structured with headings.

The `-f` option specifies the output filename.

`File1.html` and `file2.html` are the files you want to convert.

HTMLDOC will build a table of contents for the book using the heading elements (`H1`, `H2`, etc.) in your input files. It will also add a title page using the document `TITLE` text and other `META` information you supply in your files. See Chapter 4 - HTML Reference for more information on the `META` variables that are supported.

> **Note:** When using book mode, HTMLDOC starts rendering with the first `H1` element. Any text, images, tables, and other viewable elements that precede the first `H1` element are silently ignored. Because of this, make sure you have an `H1` element in your HTML file, otherwise HTMLDOC will not convert anything.

Setting the Title File

The `--titlefile` option sets the HTML, Markdown, or image file 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 or Markdown text you supply for the title page(s).

Using HTMLDOC on a Web Server

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. Filenames with spaces must usually be enclosed with quotes.

CGI Mode

HTMLDOC 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
```

The first book file that is found will be used.

Configuring HTMLDOC with Apache

The Apache web server is easily configured to use HTMLDOC. The simplest way is to copy or symlink the *htmldoc* executable to the configured *cgi-bin* directory. For example, if your Apache installation is configured to look for CGI programs in the */var/www/cgi-bin* directory, the default for Apache on Red Hat Linux, then the command to install HTMLDOC on your web server would be:

```
ln -s /usr/bin/htmldoc /var/www/cgi-bin ENTER
```

If you are using Apache 2.0.30 or higher, you will also need to enable `PATH_INFO` support by adding the following line to your *httpd.conf* file:

```
AcceptPathInfo On
```

Apache also allows you to associate CGI programs with a specific extension. If you add the following line to your *httpd.conf* file:

```
AddHandler cgi-script .cgi
```

and enable CGI execution with the `Options` directive for a directory:

```
Options +ExecCGI
```

then you can copy or symlink the *htmldoc* executable to an alternate location. For example, if you have a web directory called */var/www/htdocs/products*, you can install HTMLDOC in this directory with the following command:

```
ln -s /usr/bin/htmldoc /var/www/htdocs/products/htmldoc.cgi ENTER
```

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:

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 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
2. Choose *New->Virtual Directory* from the *Action* menu
3. Click *Next* when the *Virtual Directory Creation Wizard* window appears
4. Enter the name `htmldoc` in the *Alias* field and click *Next*
5. Enter the HTMLDOC program folder in the *Directory* field and click *Next*
6. Check the *Execute (such as ISAPI applications or CGI)* box and click *Next*
7. Click *Finish* to dismiss the wizard
8. Click on *Web Service Extensions*
9. Click *Add a new Web Service Extension*
10. Enter the name "HTMLDOC" when the *Web Service Extension* window appears
11. Click *Add...* and choose the *htmldoc.exe* file from the program folder, typically *C:\Program Files\msweet.org\HTMLDOC*
12. Check the *Set extension status to Allowed* box
13. Click *OK* to add the extension and dismiss the window

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

Once configured, the *htmldoc.exe* program will be available in the web server directory. For example, for a virtual directory called *cgi-bin*, the PDF converted URL for the *superproducts.html* page would be as follows:

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

The boldface portion represents the location of the HTMLDOC program on the web server.

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.example.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.example.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 3 - 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 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("HTMLDOC_NOCGI=1");

    # Write the content type to the client...
    header("Content-Type: application/pdf");
    flush();

    # Run HTMLDOC to provide the PDF file to the user...
    passthru("htmldoc -t pdf --quiet --jpeg --webpage $options " . escapeshellarg($filename)
}
```

The function accepts a filename and an optional "options" string for specifying the header, footer, fonts, etc.

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";
}

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 .1. " + 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());
    }
    catch (Exception e)
    {
      // An error occurred - send it to stderr for the web server...
      System.err.print(e.toString() + " caught while running:\n\n");
      System.err.print("    " + command + "\n");
      return (1);
    }
  }

  // Main entry for htmldoc class
  public static void main(String[] args)// I - Command-line args
  {
    String server_name,                 // SERVER_NAME env var
           server_port,                 // SERVER_PORT env var
           path_info,                   // PATH_INFO env var
           query_string,                // QUERY_STRING env var
           filename;                    // File to convert


    if ((server_name = System.getProperty("SERVER_NAME")) != null &&
```

```
        (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 3 - 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.md ENTER
% htmldoc options filename.book ENTER
```

The first form converts the named HTML or Markdown 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 EPUB or PDF output formats.

-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
epub	Generate an EPUB file.
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 - 1.4).
pdf11	Generate a PDF 1.1 file for Acrobat Reader 2.0 and later.
pdf12	Generate a PDF 1.2 file for Acrobat Reader 3.0 and later.
pdf13	Generate a PDF 1.3 file for Acrobat Reader 4.0 and later.
pdf14	Generate a PDF 1.4 file for Acrobat Reader 5.0 and later.
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 `-v` option specifies that progress information should be sent/displayed to the standard error file.

--batch filename.book

The `--batch` option specifies a book file that you would like to generate without the GUI popping up. This option can be combined with other options to generate the same book in different formats and sizes:

```
% htmldoc --batch filename.book -f filename.ps ENTER
% htmldoc --batch filename.book -f filename.pdf ENTER
```

--bodycolor color

The `--bodycolor` option specifies the background color for all pages in the document. The color can be specified by a standard HTML color name or as a 6-digit hexadecimal number of the form `#RRGGBB`.

--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:

typeface	Actual Font
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 = 1/72nd inch); the suffixes "in", "cm", and "mm" specify inches, centimeters, and millimeters, respectively.

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

--browserwidth pixels

The `--browserwidth` option specifies the browser width in pixels. The browser width is used to scale images and pixel measurements when generating PostScript and PDF files. It does not affect the font size of text.

The default browser width is 680 pixels which corresponds roughly to a 96 DPI display. Please note that your images and table sizes are equal to or smaller than the browser width, or your output will overlap or truncate in places.

--charset charset

The `--charset` option specifies the 8-bit character set encoding to use for the entire document. HTMLDOC comes with the following character set files:

charset	Character Set
cp-874	Windows code page 874
cp-1250	Windows code page 1250
cp-1251	Windows code page 1251
cp-1252	Windows code page 1252
cp-1253	Windows code page 1253
cp-1254	Windows code page 1254
cp-1255	Windows code page 1255
cp-1256	Windows code page 1256
cp-1257	Windows code page 1257
cp-1258	Windows code page 1258
iso-8859-1	ISO-8859-1
iso-8859-2	ISO-8859-2
iso-8859-3	ISO-8859-3
iso-8859-4	ISO-8859-4
iso-8859-5	ISO-8859-5
iso-8859-6	ISO-8859-6
iso-8859-7	ISO-8859-7
iso-8859-8	ISO-8859-8
iso-8859-9	ISO-8859-9
iso-8859-14	ISO-8859-14
iso-8859-15	ISO-8859-15
koi8-r	KOI8-R
utf-8	UTF-8

Note: UTF-8 support is limited to the first 128 Unicode characters found in the input.

--color

The `--color` option specifies that color output is desired.

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

--compression[=level]

The `--compression` option specifies that Flate compression should be performed on the output file(s). The optional `level` parameter is a number from 1 (fastest and least amount of compression) to 9 (slowest and most amount of compression).

This option is only available when generating PDF or Level 3 PostScript files.

--continuous

The `--continuous` option specifies that the input files comprise a web page (or site) and that no title page or table-of-contents should be generated. Unlike the `--webpage` option described later in this chapter, page breaks are not inserted between each input file.

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

--cookies 'name=\"value with space\"; name=value'

The `--cookies` option specifies one or more HTTP cookies that should be sent when converting remote URLs. Each cookie must be separated from the others by a semicolon and a space, and values containing whitespace or the semicolon must be placed inside double-quotes. When specifying multiple cookies, the entire cookie string must be surrounded by single quotes in order for the string to be processed correctly.

--datadir directory

The `--datadir` option specifies the location of data files used by HTMLDOC.

--duplex

The `--duplex` option specifies that the output should be formatted for two sided printing.

This option is only available when generating PostScript or PDF files. Use the `--pscommands` option to generate PostScript duplex mode commands.

--effectduration seconds

The `--effectduration` option specifies the duration of a page transition effect in seconds.

This option is only available when generating PDF files.

--embedfonts

The `--embedfonts` option specifies that fonts should be embedded in PostScript and PDF output. This is especially useful when generating documents in character sets other than ISO-8859-1.

--encryption

The `--encryption` option enables encryption and security features for PDF output.

This option is only available when generating PDF files.

--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:

page	Description
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.

--footer lcr

The `--footer` option specifies the contents of the page footer. The `lcr` parameter is a three-character string representing the left, center, and right footer fields. Each character can be one of the following:

lcr	Description
.	A period indicates that the field should be blank.
:	A colon indicates that the field should contain the current and total number of pages in the chapter (n/N).
/	A slash indicates that the field should contain the current and total number of pages (n/N).
1	The number 1 indicates that the field should contain the current page number in decimal format (1, 2, 3, ...)
a	A lowercase "a" indicates that the field should contain the current page number using lowercase letters.
A	An uppercase "A" indicates that the field should contain the current page number using UPPERCASE letters.
c	A lowercase "c" indicates that the field should contain the current chapter title.
C	An uppercase "C" indicates that the field should contain the current chapter page number.
d	A lowercase "d" indicates that the field should contain the current date.
D	An uppercase "D" indicates that the field should contain the current date and time.
h	An "h" indicates that the field should contain the current heading.
i	A lowercase "i" indicates that the field should contain the current page number in lowercase roman numerals (i, ii, iii, ...)
I	An uppercase "I" indicates that the field should contain the current page number in uppercase roman numerals (I, II, III, ...)
l	A lowercase "l" indicates that the field should contain the logo image.
l	An uppercase "L" indicates that the field should contain the logo image as a letterhead (shown at full size).
t	A lowercase "t" indicates that the field should contain the document title.
T	An uppercase "T" indicates that the field should contain the current time.
u	A lowercase "u" indicates that the field should contain the current filename or URL.

Setting the footer to "`...`" disables the footer entirely.

--format format

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

Format	Description
epub	Generate an EPUB file.
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 - 1.4).
pdf11	Generate a PDF 1.1 file for Acrobat Reader 2.0 and later.
pdf12	Generate a PDF 1.2 file for Acrobat Reader 3.0 and later.
pdf13	Generate a PDF 1.3 file for Acrobat Reader 4.0 and later.
pdf14	Generate a PDF 1.4 file for Acrobat Reader 5.0 and later.
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.

--header lcr

The `--header` option specifies the contents of the page header. The `lcr` parameter is a three-character string representing the left, center, and right header fields. See the `--footer` option for the list of formatting characters.

Setting the header to "`...`" disables the header entirely.

--header1 lcr

The `--header1` option specifies the contents of the page header for the first body/chapter page. The `lcr` parameter is a three-character string representing the left, center, and right header fields. See the `--footer` option for the list of formatting characters.

Setting the header to "`...`" disables the first page header entirely.

--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.

--headingfont typeface

The `--headingfont` options sets the typeface that is used for headings in the document. The `typeface` parameter can be one of the following:

typeface	Actual Font
Arial	Helvetica
Courier	Courier
Helvetica	Helvetica
Monospace	DejaVu Sans Mono
Sans	DevaVu Sans
Serif	DejaVu Serif
Times	Times

--help

The `--help` option displays all of the available options to the standard output file.

--helpdir directory

The `--helpdir` option specifies the location of the on-line help files.

--hfimage*N* filename

The `--hfimage`*N* option specifies an image to use in the header and/or footer, where N is a number from 1 to 10. The supported formats are BMP, GIF, JPEG, and PNG.

--jpeg[=quality]

The `--jpeg` option enables JPEG compression of continuous-tone images. The optional `quality` parameter specifies the output quality from 0 (worst) to 100 (best).

This option is only available when generating PDF or Level 2 and Level 3 PostScript files.

--landscape

The `--landscape` option specifies that the output should be in landscape orientation (long edge on top).

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

--left margin

The `--left` option specifies the left margin. The default units are points (1 point = 1/72nd inch); the suffixes "in", "cm", and "mm" specify inches, centimeters, and millimeters, respectively.

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

--letterhead filename

The `--letterhead` option specifies the letterhead image for the page headers and footers for PostScript and PDF files. The supported formats are BMP, GIF, JPEG, and PNG.

> **Note:** You need to use the `--header` and/or `--footer` options with the `L` parameter or use the corresponding HTML page comments to display the logo image in the header or footer.

--linkcolor color

The `--linkcolor` option specifies the color of links in EPUB, HTML. and PDF output. The color can be specified by name or as a 6-digit hexadecimal number of the form `#RRGGBB`.

--links

The `--links` option specifies that PDF output should contain hyperlinks.

--linkstyle style

The `--linkstyle` option specifies the style of links in EPUB, HTML, and PDF output. The style can be "plain" for no decoration or "underline" to underline links.

--logoimage filename

The `--logoimage` option specifies the logo image for the HTML navigation bar and page headers and footers for PostScript and PDF files. The supported formats are BMP, GIF, JPEG, and PNG.

> **Note:** You need to use the `--header` and/or `--footer` options with the `l` parameter or use the corresponding HTML page comments to display the logo image in the header or footer.

--no-compression

The `--no-compression` option specifies that Flate compression should not be performed on the output files.

--no-duplex

The `--no-duplex` option specifies that the output should be formatted for one sided printing.

This option is only available when generating PostScript or PDF files. Use the `--pscommands` option to generate PostScript duplex mode commands.

--no-embedfonts

The `--no-embedfonts` option specifies that fonts should not be embedded in PostScript and PDF output.

--no-encryption

The `--no-encryption` option specifies that no encryption/security features should be enabled in PDF output.

This option is only available when generating PDF files.

--no-jpeg

The `--no-jpeg` option specifies that JPEG compression should not be performed on large images.

--no-links

The `--no-links` option specifies that PDF output should not contain hyperlinks.

--no-localfiles

The `--no-localfiles` option disables access to local files on the system. This option should be used when providing remote document conversion services.

--no-numbered

The `--no-numbered` option specifies that headings should not be numbered.

--no-pscommands

The `--no-pscommands` option specifies that PostScript device commands should not be written to the output files.

--no-strict

The `--no-strict` option turns off strict HTML conformance checking.

--no-title

The `--no-title` option specifies that the title page should not be generated.

--no-toc

The `--no-toc` option specifies that the table-of-contents pages should not be generated.

--no-xrxcomments

The `--no-xrxcomments` option specifies that Xerox PostScript job comments should not be written to the output files.

This option is only available when generating PostScript files.

--numbered

The `--numbered` option specifies that headings should be numbered.

--nup pages

The `--nup` option sets the number of pages that are placed on each output page. Valid values for the `pages` parameter are 1, 2, 4, 6, 9, and 16.

--outdir directory

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

This option is not compatible with the PDF output format.

--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.

--pageeffect effect

The `--pageeffect` option specifies the page effect to use in PDF files. The `effect` parameter can be one of the following:

effect	Description
none	No effect is generated.
bi	Box Inward
bo	Box Outward
d	Dissolve
gd	Glitter Down
gdr	Glitter Down and Right
gr	Glitter Right
hb	Horizontal Blinds
hsi	Horizontal Sweet Inward
hso	Horizontal Sweep Outward
vb	Vertical Blinds
vsi	Vertical Sweep Inward
vso	Vertical Sweep Outward
wd	Wipe Down
wl	Wipe Left
wr	Wipe Right
wu	Wipe Up

This option is only available when generating PDF files.

--pagelayout layout

The `--pagelayout` option specifies the initial page layout in the PDF viewer. The `layout` parameter can be one of the following:

layout	Description
single	A single page is displayed.
one	A single column is displayed.
twoleft	Two columns are displayed with the first page on the left.
tworight	Two columns are displayed with the first page on the right.

This option is only available when generating PDF files.

--pagemode mode

The `--pagemode` option specifies the initial viewing mode in the PDF viewer. The `mode` parameter can be one of the following:

mode	Description
document	The document pages are displayed in a normal window.
outline	The document outline and pages are displayed.
fullscreen	The document pages are displayed on the entire screen in "slideshow" mode.

This option is only available when generating PDF files.

--path dir1;dir2;dir3;...;dirN

The `--path` option 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 URLs can be specified. Quotes around the directory parameter are optional. They are usually used when the directory string contains spaces.

```
--path "dir1;dir2;dir3;...;dirN"
```

--permissions permission[,permission,...]

The `--permissions` option specifies the document permissions. The available permission parameters are listed below:

Permission	Description
all	All permissions
annotate	User can annotate document
copy	User can copy text and images from document
modify	User can modify document
print	User can print document
no-annotate	User cannot annotate document
no-copy	User cannot copy text and images from document
no-modify	User cannot modify document
no-print	User cannot print document
none	No permissions

The `--encryption` option must be used in conjunction with the `--permissions` parameter.

```
--permissions no-print --encryption
```

Multiple options can be specified by separating them with commas:

```
--permissions no-print,no-copy --encryption
```

This option is only available when generating PDF files.

--portrait

The `--portrait` option specifies that the output should be in portrait orientation (short edge on top).

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

--pscommands

The `--pscommands` option specifies that PostScript device commands should be written to the output files.

This option is only available when generating Level 2 and Level 3 PostScript files.

--quiet

The `--quiet` option prevents error messages from being sent to stderr.

--referer url

The `--referer` option sets the URL that is passed in the `Referer:` field of HTTP requests.

--right margin

The `--right` option specifies the right margin. The default units are points (1 point = 1/72nd inch); the suffixes "in", "cm", and "mm" specify inches, centimeters, and millimeters, respectively.

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

--size size

The `--size` option specifies the page size. The `size` parameter can be one of the following standard sizes:

size	Description
Letter	8.5x11in (216x279mm)
A4	8.27x11.69in (210x297mm)
Universal	8.27x11in (210x279mm)

Custom sizes are specified by the page width and length separated by the letter "x" to select a custom page size. Append the letters "in" for inches, "mm" for millimeters, or "cm" for centimeters.

This option is only available when generating PostScript or PDF files. Use the `--pscommands` option to generate PostScript page size commands.

--strict

The `--strict` option turns on strict HTML conformance checking. When enabled, HTML elements that are improperly nested and dangling close elements will produce error messages.

--textcolor color

The `--textcolor` option specifies the default text color for all pages in the document. The color can be specified by a standard HTML color name or as a 6-digit hexadecimal number of the form `#RRGGBB`.

--textfont typeface

The `--textfont` options sets the typeface that is used for text in the document. The `typeface` parameter can be one of the following:

typeface	Actual Font
Arial	Helvetica
Courier	Courier
Helvetica	Helvetica
Monospace	DejaVu Sans Mono
Sans	DevaVu Sans
Serif	DejaVu Serif
Times	Times

--title

The `--title` option specifies that a title page should be generated.

--titlefile filename

The `--titlefile` option specifies a HTML or Markdown file to use for the title page.

--titleimage filename

The `--titleimage` option specifies the title image for the title page. The supported formats are BMP, GIF, JPEG, and PNG.

--tocfooter lcr

The `--tocfooter` option specifies the contents of the table-of-contents footer. The `lcr` parameter is a three-character string representing the left, center, and right footer fields. See the `--footer` option for the list of formatting characters.

Setting the TOC footer to "`...`" disables the TOC footer entirely.

--tocheader lcr

The `--tocheader` option specifies the contents of the table-of-contents header. The `lcr` parameter is a three-character string representing the left, center, and right header fields. See the `--footer` option for the list of formatting characters.

Setting the TOC header to "`...`" disables the TOC header entirely.

--toclevels levels

The `--toclevels` options specifies the number of heading levels to include in the table-of-contents pages. The `levels` parameter is a number from 1 to 6.

--toctitle string

The `--toctitle` options specifies the string to display at the top of the table-of-contents; the default string is "Table of Contents".

--top margin

The `--top` option specifies the top margin. The default units are points (1 point = 1/72nd inch); the suffixes "in", "cm", and "mm" specify inches, centimeters, and millimeters, respectively.

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

--user-password password

The `--user-password` option specifies the user password for a PDF file. If not specified or the empty string (""), no password will be required to view the document.

This option is only available when generating PDF files.

--verbose

The `--verbose` option specifies that progress information should be sent/displayed to the standard error file.

--version

The `--version` option displays the HTMLDOC version number.

--webpage

The `--webpage` option specifies that the input files comprise a web page (or site) and that no title page or table-of-contents should be generated. HTMLDOC will insert a page break between each input file.

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

--xrxcomments

The `--xrxcomments` option specifies that Xerox PostScript job comments should be written to the output files.

This option is only available when generating PostScript files.

Environment Variables

HTMLDOC looks for several environment variables which can override the default directories, display additional debugging information, and disable CGI mode.

HTMLDOC_DATA

This environment variable specifies the location of HTMLDOC's *data* and *fonts* directories, normally */usr/share/htmldoc* or *C:\Program Files\HTMLDOC*.

HTMLDOC_DEBUG

This environment variable enables debugging information that is sent to stderr. The value is a list of keywords separated by spaces:

keyword	Information Shown
`links`	Shows all of the links in a document
`memory`	Shows memory usage statistics
`remotebytes`	Shows the number of bytes that were transferred via HTTP
`table`	Puts a box around each table, row, and cell
`tempfiles`	Shows the temporary files that were created, and preserves them for debugging
`timing`	Shows the load and render times
`all`	All of the above

HTMLDOC_HELP

This environment variable specifies the location of HTMLDOC's documentation directory, normally */usr/share/doc/htmldoc* or *C:\Program Files\HTMLDOC\doc*.

HTMLDOC_NOCGI

This environment variable, when set (the value doesn't matter), disables CGI mode. It is most useful for using HTMLDOC on a web server from a scripting language or invocation from a program.

Messages

HTMLDOC sends error and status messages to stderr unless the `--quiet` option is provided on the command-line. Applications can capture these messages to relay errors or statistics to the user.

BYTES: Message

The `BYTES:` message specifies the number of bytes that were written to an output file. If the output is directed at a directory then multiple `BYTES:` messages will be sent.

DEBUG: Messages

The `DEBUG:` messages contain debugging information based on the value of the `HTMLDOC_DEBUG` environment variable. Normally, no `DEBUG:` messages are sent by HTMLDOC.

ERRnnn: Messages

The `ERRnnn:` messages specify an error condition. Error numbers 1 to 14 map to the following errors:

1. No files were found or loadable.
2. No pages were generated.
3. The document contains too many files or chapters.
4. HTMLDOC ran out of memory.
5. The specified file could not be found.
6. The comment contains a bad HTMLDOC formatting command.
7. The image file is not in a known format.
8. HTMLDOC was unable to remove a temporary file.
9. HTMLDOC had an unspecified internal error.
10. HTMLDOC encountered a networking error when retrieving a file via a URL.
11. HTMLDOC was unable to read a file.
12. HTMLDOC was unable to write a file.
13. A HTML error was found in a source file.
14. A table, image, or text fragment was too large to fit in the space provided.
15. A hyperlink in the source files was unresolved.
16. A header/footer string in the document contains a bad $ command.

Error numbers 100 to 505 correspond directly to a HTTP status code.

INFO: Messages

The `INFO:` messages contain general information that is logged when HTMLDOC is running in CGI mode or when you use the `--verbose` option.

PAGES: Message

The `PAGES:` message specifies the number of pages that were written to an output file. If the output is directed at a directory then multiple `PAGES:` messages will be sent. No `PAGES:` messages are sent when generating HTML or EPUB output.

REMOTEBYTES: Message

The `REMOTEBYTES:` message specifies the number of bytes that were transferred using HTTP. This message is only displayed if the `HTMLDOC_DEBUG` environment variable has the keyword `remotebytes` or `all`.

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`.

Chapter 4 - 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	See Below
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	See Below
FORM	2.0	No	
FRAME	3.2	No	

Element	Version	Supported?	Notes
FRAMESET	3.2	No	
H1	1.0	Yes	Boldface, See Below
H2	1.0	Yes	Boldface, See Below
H3	1.0	Yes	Boldface, See Below
H4	1.0	Yes	Boldface, See Below
H5	1.0	Yes	Boldface, See Below
H6	1.0	Yes	Boldface, See Below
HEAD	1.0	Yes	
HR	1.0	Yes	
HTML	1.0	Yes	
I	1.0	Yes	
IMG	1.0	Yes	See Below
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	See Below
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	See Below
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	

Comments

HTMLDOC supports many special HTML comments to initiate page breaks, set the header and footer text, and control the current media options:

<!-- FOOTER LEFT "foo" -->
: Sets the left footer text; the test is applied to the current page if empty, or the next page otherwise.

<!-- FOOTER CENTER "foo" -->
: Sets the center footer text; the test is applied to the current page if empty, or the next page otherwise.

<!-- FOOTER RIGHT "foo" -->
: Sets the right footer text; the test is applied to the current page if empty, or the next page otherwise.

<!-- HALF PAGE -->
: Break to the next half page.

<!-- HEADER LEFT "foo" -->
: Sets the left header text; the test is applied to the current page if empty, or the next page otherwise.

<!-- HEADER CENTER "foo" -->
: Sets the center header text; the test is applied to the current page if empty, or the next page otherwise.

`<!-- HEADER RIGHT "foo" -->`
: Sets the right header text; the test is applied to the current page if empty, or the next page otherwise.

`<!-- MEDIA BOTTOM nnn -->`
: Sets the bottom margin of the page. The "nnn" string can be any standard measurement value, e.g. 0.5in, 36, 12mm, etc. Breaks to a new page if the current page is already marked.

`<!-- MEDIA COLOR "foo" -->`
: Sets the media color attribute for the page. The "foo" string is any color name that is supported by the printer, e.g. "Blue", "White", etc. Breaks to a new page or sheet if the current page is already marked.

`<!-- MEDIA DUPLEX NO -->`
: Chooses single-sided printing for the page; breaks to a new page or sheet if the current page is already marked.

`<!-- MEDIA DUPLEX YES -->`
: Chooses double-sided printing for the page; breaks to a new sheet if the current page is already marked.

`<!-- MEDIA LANDSCAPE NO -->`
: Chooses portrait orientation for the page; breaks to a new page if the current page is already marked.

`<!-- MEDIA LANDSCAPE YES -->`
: Chooses landscape orientation for the page; breaks to a new page if the current page is already marked.

`<!-- MEDIA LEFT nnn -->`
: Sets the left margin of the page. The "nnn" string can be any standard measurement value, e.g. 0.5in, 36, 12mm, etc. Breaks to a new page if the current page is already marked.

`<!-- MEDIA POSITION nnn -->`
: Sets the media position attribute (input tray) for the page. The "nnn" string is an integer that usually specifies the tray number. Breaks to a new page or sheet if the current page is already marked.

`<!-- MEDIA RIGHT nnn -->`
: Sets the right margin of the page. The "nnn" string can be any standard measurement value, e.g. 0.5in, 36, 12mm, etc. Breaks to a new page if the current page is already marked.

`<!-- MEDIA SIZE foo -->`
: Sets the media size to the specified size. The "foo" string can be "Letter", "Legal", "Universal", or "A4" for standard sizes or "WIDTHxHEIGHTunits" for custom sizes, e.g. "8.5x11in"; breaks to a new page or sheet if the current page is already marked.

`<!-- MEDIA TOP nnn -->`
: Sets the top margin of the page. The "nnn" string can be any standard measurement value, e.g. 0.5in, 36, 12mm, etc. Breaks to a new page if the current page is already marked.

`<!-- MEDIA TYPE "foo" -->`
: Sets the media type attribute for the page. The "foo" string is any type name that is supported by the printer, e.g. "Plain", "Glossy", etc. Breaks to a new page or sheet if the current page is already marked.

`<!-- NEED length -->`
: Break if there is less than `length` units left on the current page. The `length` value defaults to lines of text but can be suffixed by `in`, `mm`, or `cm` to convert from the corresponding units.

`<!-- NEW PAGE -->`
: Break to the next page.

`<!-- NEW SHEET -->`
: Break to the next sheet.

`<!-- NUMBER-UP nn -->`
: Sets the number of pages that are placed on each output page. Valid values are 1, 2, 4, 6, 9, and 16.

`<!-- PAGE BREAK -->`
: Break to the next page.

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.

`$HFIMAGE1`
`$HFIMAGE2`
`$HFIMAGE3`
`$HFIMAGE4`
`$HFIMAGE5`
`$HFIMAGE6`
`$HFIMAGE7`
`$HFIMAGE8`
`$HFIMAGE9`
`$HFIMAGE10`
: Inserts the specified header/footer image; all other text in the string will be ignored.

`$LETTERHEAD`
: Inserts the logo image as a letterhead with no down-scaling; all other text in the string will be ignored.

`$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 the current time.

`$TITLE`
: Inserts the document title.

`$URL`
: Inserts the document filename or URL.

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:** Heading levels 7 to 15 are not standard HTML and will not likely be recognized by most web browsers.

The headings you use within a chapter must start at level 2 (H2). If you skip levels the heading will be shown under the last level that was known. For example, if you use the following hierarchy of headings:

```
<H1>Chapter Heading</H1>
...
<H2>Section Heading 1</H2>
...
<H2>Section Heading 2</H2>
...
<H3>Sub-Section Heading 1</H3>
...
<H4>Sub-Sub-Section Heading 1</H4>
...
<H4>Sub-Sub-Section Heading 2</H4>
...
<H3>Sub-Section Heading 2</H3>
```

```
...
<H2>Section Heading 3</H2>
...
<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"`
: Specifies that decimal numbers should be generated for this heading level.

`TYPE="a"`
: Specifies that lowercase letters should be generated for this heading level.

`TYPE="A"`
: Specifies that uppercase letters should be generated for this heading level.

`TYPE="i"`
: Specifies that lowercase roman numerals should be generated for this heading level.

`TYPE="I"`
: Specifies that uppercase roman numerals should be generated for this heading level.

Images

HTMLDOC supports loading of BMP, GIF, JPEG, and PNG image files. EPS and other types of image files are not supported at this time.

Links

External URL and internal (`#target` and `filename.html`) links are fully supported for HTML and PDF output.

When generating PDF files, local PDF file links will be converted to external file links for the PDF viewer instead of URL links. That is, you can directly link to another local PDF file from your HTML document with:

```
<A HREF="filename.pdf">...</A>
```

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="HTMLDOC.filename" CONTENT="..."`
: Specifies the filename that is reported in CGI mode.

`<META NAME="KEYWORDS" CONTENT="..."`
: Specifies document search keywords.

`<META NAME="SUBJECT" CONTENT="..."`
: Specifies document subject.

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 code.

HTMLDOC does not support HTML 4.0 table elements or attributes, such as `TBODY`, `THEAD`, `TFOOT`, or `RULES`.

Chapter 5 - Markdown Reference

This chapter describes the markdown syntax that is recognized and supported by HTMLDOC.

General Syntax

Markdown is a simple plain-text format that uses formatting conventions that are commonly used in email and other text-based communications. Markdown is used by most of the major blogging, web site, and project hosting platforms and is supported by many standalone text editors.

HTMLDOC supports the CommonMark version of markdown syntax with the following exceptions:

- Metadata as used by Jekyll and other web markdown solutions can be placed at the beginning of the file;
- "@" links can be used which resolve to headings within the file;
- Tables can be embedded using the "|" separator;
- Embedded HTML markup and entities are explicitly not supported or allowed;
- Tabs are silently expanded to the markdown standard of four spaces since HTML uses eight spaces per tab; and
- Some pathological nested link and inline style features supported by CommonMark (`******Really Strong Text******`) are not supported by `mmd`.

  Note: HTMLDOC does not support embedded HTML in markdown documents because the version of HTML (or XHTML) cannot be reliably determined, making support of certain character entities and language elements problematic.

Metadata Syntax

Metadata is specified at the top of a markdown file between two lines containing three hyphens, for example:

```
---
title: My Great Novel
author: John Doe
copyright: Copyright © 2018 by John Doe
version: 1.0
language: en-US
subject: Fiction
---

# Preamble

...
```

HTMLDOC supports the "author", "copyright", "language", "subject", "title", and "version" metadata and silently ignores everything else.

Link Targets and @ Links

CommonMark defines no standard for how implementations generate anchors or identifiers for headings in a markdown file - this makes hyperlinking to a named section within a document basically impossible. Jekyll and other markdown implementations allow the special link "@" to be used, which HTMLDOC supports:

```
See [Screwing in a Light Bulb](@) for instructions on installing a
light bulb.

...

# Screwing in a Light Bulb

...
```

To reference a markdown heading from a HTML file, convert the heading to lowercase, replace spaces with the hyphen ("-"), and remove any special characters. Thus, a HTML file would reference the previous heading using the following HTML:

```
<a href="#screwing-in-a-light-bulb"> ... </a>
```

Table Syntax

CommonMark does not define a syntax for plain-text tables, instead relying on embedded HTML which HTMLDOC does not support. Both Github and Jekyll support a common markdown extension for plain text tables that uses the vertical pipe ("|") character to specify column separations. The first line contains the table header, the second line is a horizontal separator, and the remaining lines contain the table body. For example:

```
| Heading 1 | Heading 2 | Heading 3 |
| --------- | --------- | --------- |
| Cell 1,1  | Cell 1,2  | Cell 1,3  |
| Cell 2,1  | Cell 2,2  | Cell 2,3  |
| Cell 3,1  | Cell 3,2  | Cell 3,3  |
```

will produce:

Heading 1	Heading 2	Heading 3
Cell 1,1	Cell 1,2	Cell 1,3
Cell 2,1	Cell 2,2	Cell 2,3
Cell 3,1	Cell 3,2	Cell 3,3

The outer pipes can be omitted, for example:

```
Heading 1 | Heading 2 | Heading 3
--------- | --------- | ---------
Cell 1,1  | Cell 1,2  | Cell 1,3
Cell 2,1  | Cell 2,2  | Cell 2,3
Cell 3,1  | Cell 3,2  | Cell 3,3
```

While table headings are always centered, you can control the alignment of the body cells by using the colon (":") character in the separator line. Put a leading colon to specify left alignment (the default), a trailing colon for right alignment, or both to specify centering. For example:

```
Left Alignment | Center Alignment | Right Alignment
:------------- | :--------------: | --------------:
Cell 1,1       |     Cell 1,2     |               1
Cell 2,1       |     Cell 2,2     |              12
Cell 3,1       |     Cell 3,2     |             123
```

will produce:

Left Alignment	Center Alignment	Right Alignment
Cell 1,1	Cell 1,2	1
Cell 2,1	Cell 2,2	12
Cell 3,1	Cell 3,2	123

Table columns do not need to be padded so that they line up - the following (less readable) example is perfectly valid:

```
Left Alignment|Center Alignment|Right Alignment
:--|:--:|--:
Cell 1,1|Cell 1,2|1
Cell 2,1|Cell 2,2|12
Cell 3,1|Cell 3,2|123
```

Appendix A - License Agreement

GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

```
Copyright 1989, 1991 Free Software Foundation, Inc.
59 Temple Place, Suite 330, Boston, MA 02111-1307 USA

Everyone is permitted to copy and distribute verbatim
copies of this license document, but changing it is not allowed.
```

Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

GNU GENERAL PUBLIC LICENSE
TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

   Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the Program (independent of having been made by running the Program). Whether that is true depends on what the Program does.
1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

   You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.
2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:
   a. You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.
   b. You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.
   c. if the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this License. (Exception: if

the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:
    a. Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
    b. Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
    c. Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

   The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

   If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.
4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.
5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.
7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

   If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to apply and the section as a whole is intended to apply in other circumstances.

   It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

   This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.
8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.
9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

   Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.
10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY MODIFY AND/OR REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

How to Apply These Terms to Your New Programs

If you develop a new program, and you want it to be of the greatest possible use to the public, the best way to achieve this is to make it free software which everyone can redistribute and change under these terms.

To do so, attach the following notices to the program. It is safest to attach them to the start of each source file to most effectively convey the exclusion of warranty; and each file should have at least the "copyright" line and a pointer to where the full notice is found.

```
one line to give the program's name and an idea of what it does.
Copyright (C) yyyy name of author

This program is free software; you can redistribute it and/or
modify it under the terms of the GNU General Public License
as published by the Free Software Foundation; either version 2
of the License, or (at your option) any later version.

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
GNU General Public License for more details.

You should have received a copy of the GNU General Public License
along with this program; if not, write to the Free Software
Foundation, Inc., 59 Temple Place - Suite 330, Boston, MA  02111-1307, USA.
```

Also add information on how to contact you by electronic and paper mail.

If the program is interactive, make it output a short notice like this when it starts in an interactive mode:

```
Gnomovision version 69, Copyright (C) yeaname of author
Gnomovision comes with ABSOLUTELY NO WARRANTY; for details
type `show w'.  This is free software, and you are welcome
to redistribute it under certain conditions; type `show c'
for details.
```

The hypothetical commands `` `show w' `` and `` `show c' `` should show the appropriate parts of the General Public License. Of course, the commands you use may be called something other than `` `show w' `` and `` `show c' ``; they could even be mouse-clicks or menu items--whatever suits your program.

You should also get your employer (if you work as a programmer) or your school, if any, to sign a "copyright disclaimer" for the program, if necessary. Here is a sample; alter the names:

```
Yoyodyne, Inc., hereby disclaims all copyright
```

```
interest in the program `Gnomovision'
(which makes passes at compilers) written
by James Hacker.
```

signature of Ty Coon, 1 April 1989
Ty Coon, President of Vice

Appendix B - Book File Format

This appendix describes the HTMLDOC *.book* file format.

Introduction

The HTMLDOC *.book* file format is a simple text format that provides the command-line options and files that are part of the document. These files can be used from the GUI interface or from the command-line using the `--batch` option:

```
htmldoc filename.book
htmldoc --batch filename.book
```

The first form will load the book and display the GUI interface, if configured. Windows users should use *ghtmldoc.exe* executable to show the GUI and *htmldoc.exe* for the batch mode:

```
ghtmldoc.exe filename.book
htmldoc.exe --batch filename.book
```

The Header

Each *.book* file starts with a line reading:

```
#HTMLDOC 1.9
```

The version number (1.9) is optional.

The Options

Following the header is a line containing the options for the book. You can use any valid command-line option on this line:

```
-f htmldoc.pdf --titleimage htmldoc.png --duplex --compression=9 --jpeg=90
```

Long option lines can be broken using a trailing backslash (\) on the end of each continuation line:

```
-f htmldoc.pdf --titleimage htmldoc.png --duplex \
--compression=9 --jpeg=90
```

The Files

Following the options are a list of files or URLs to include in the document:

```
1-intro.html
2-using.html
3-cmdref.html
4-htmlref.html
5-mdref.html
a-license.html
b-book.html
```

Putting It All Together

The following is the complete book file needed to generate this documentation:

```
#HTMLDOC 1.9
-f htmldoc.pdf --titleimage htmldoc.png --duplex --compression=9 --jpeg=90
1-intro.html
2-using.html
3-cmdref.html
4-htmlref.html
5-mdref.html
a-license.html
b-book.html
```