HTMLDOC Users Manual

Michael R Sweet

Table of Contents

Table of Contents

Table of Contents

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 opensource 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 toolswas prohibitive to my small business.

In the end I decided to write my own program to generate the documentation. HTML seemed to be thesource 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 printfrom 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.

HTMLDOCused to be available under a commercial end-user license agreementfrom 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:

Encryption Support

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

Legal Stuff

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 afile to be converted, click on the *Output* tab to set the outputfile 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 intoPostScript 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 isused 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 markedusing the H2 through H6 elements.

> **Note:** Whenusing 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 titlefile is HTML or Markdown, the contents are formatted to producetitle page(s). When thetitlefile is an image, the image is centered on thetitle page with automatically generate content based on thetitle and other metadata.

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

Using theHTMLDOC 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 outputfile name (`output.pdf`). In this example it is a PDF file.

`Filename.html` is the name of thefile 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 beenusing 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 bookfrom 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 bookusing the heading elements (`H1`, `H2`, etc.) in your input files. It will also add atitle pageusing the document `TITLE` text and other `META` information you supply in yourfiles. See Chapter 4 - HTML Reference for more information on the `META` variables that are supported.

> **Note:** Whenusing 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 silentlyignored. Because of this, make sure you have an `H1` element inyour HTML file, otherwiseHTMLDOC will not convert anything.

Setting the Title File

The `--titlefile` option sets the HTML, Markdown, or imagefile to use on thetitle 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 GIF, JPEG, and PNG images, as well as generic HTML or Markdown text you supply for thetitle page(s).

UsingHTMLDOC on a Web Server

HTMLDOC can beused in a variety of ways to generate formatted reports on a web server. The most common way is to useHTMLDOC as a CGI program withyour web server to provide PDF-formatted output of a web page. Examples are providedfor Microsoft IIS and the Apache web servers.

HTMLDOC can also be calledfromyour own server-side scripts and programs. Examples are providedfor PHP and Java.

> **Warning:** Passing information directlyfrom the web browser toHTMLDOC can potentially exposeyour system to security risks. Always be sure to "sanitize" any inputfrom the web browser so that filenames, URLs, and options passed toHTMLDOC 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 toyour web server's *cgi-bin* directory and then use it to produce PDF

versions ofyour web pages.

The CGI converts a page onyour 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 installedHTMLDOC inyour server's *cgi-bin* directory, you would directyour clients to the following URL:

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

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

Form datausing 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 bookfile to set any preferencesfor the conversion to PDF. For the *superproducts.html*file described previously,HTMLDOC will look at the following URLsfor a book file:

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

The first bookfile that is found will be used.

ConfiguringHTMLDOC with Apache

The Apache web server is easily configuredto use HTMLDOC. The simplest way isto copyor symlink the *htmldoc* executableto the configured *cgi-bin* directory. For example, ifyour Apache installation is configuredto lookfor CGI programs in the */var/www/cgi-bin* directory, the defaultfor Apache on Red Hat Linux, then the commandto installHTMLDOC onyour web server would be:

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

If you areusing Apache 2.0.30or higher, you will also needto enable `PATH_INFO` support by adding the following linetoyour *httpd.conf* file:

```
AcceptPathInfo On
```

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

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

and enable CGI execution with the `Options` directivefor a directory:

```
Options +ExecCGI
```

then you can copyor symlink the *htmldoc* executableto an alternate location. For example, if you have a web directory called */var/www/htdocs/products*, you can installHTMLDOC in thisdirectory with the following command:

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

ConfiguringHTMLDOC with Microsoft IIS

The IIS web server isconfiguredto run CGI programs by either modifying the permissions of an existingdirectoryor by creating a new virtualdirectory that allowsfor 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 stepsto add a virtual folderfor HTMLDOC:

1. Click onyour server in the listto 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 theHTMLDOC program folder in the *Directory* field and click *Next*
6. Check the *Execute (such as ISAPI applicationsor 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*filefrom the program folder, typically *C:\Program Files\msweet.org\HTMLDOC*
12. Check the *Set extension statusto Allowed* box
13. Click *OK*to add the extension and dismiss the window

Finally, double-click the *My Computer* icon on the desktopor start the *Windows Explorer*. When the explorer window appears, perform the following stepsto provide write accessto the Windows temporary folder:

1. Open the windows temporaryfile folder, typically *C:\WINDOWS\TEMP*
2. Choose *Properties*from the *File* menu
3. Click on the *Security* tab
4. Click *Add...*, enter the usernamefor the web server, typically "SERVER\IUSR_SERVER" where "SERVER" is the name you gaveyour 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 virtualdirectory called *cgi-bin*, the PDF converted URLfor the *superproducts.html* page would be as follows:

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

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

UsingHTMLDOC From Server-Side Scripts and Programs

To make this work the CGI scriptor program must send the appropriate HTTP attributes, the required emptylineto signify the beginning of the document, and then execute theHTMLDOC programto generate the HTML, PostScript,or PDFfile as needed. SinceHTMLDOC looksfor CGI environment variables when it is run, you must also setthe `HTMLDOC_NOCGI` environment variableto a value of 1 before runningHTMLDOCfromyour CGI scriptor program.

Another wayto generate PDF filesfromyour reports isto useHTMLDOC as a "portal" application. Whenused as a portal,HTMLDOC automatically retrievesthe named documentor reportfromyour server and passes a PDF versiontothe web browser. Seethe next sectionsfor more information.

CallingHTMLDOCfrom a Shell Script

Shell scripts are probablythe easiestto work with, but are normally limitedto GET type requests. Here is a script called *topdf* that acts as a portal, convertingthe namedfileto 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 referencethe URL "http://www.example.com/topdf.cgi/index.html"to generate a PDFfile ofthe site's home page.

The *options* variable inthescript can be setto use any supported command-line optionfor HTMLDOC;for a complete list see Chapter 3 - Command-Line Reference.

CallingHTMLDOCfrom Perl

Perl scripts offerthe abilityto generate more complex reports, pull datafrom databases, etc. The easiest wayto interface Perl scripts withHTMLDOC isto write a reportto a temporaryfile and then executeHTMLDOCto generatethePDF file.

Here is a simple Perl subroutine that can beusedto write aPDF reporttothe 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";
}
```

CallingHTMLDOCfromPHP

PHP provides a `passthru()` function that can beusedto run HTMLDOC. This combined withthe `header()` function can beusedto provide on-the-fly reports inPDF format.

Here is a simplePHP function that can beusedtoconvert a HTML reporttoPDF and send ittothe 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 $options " . escapeshellarg($filena
}
```

The function accepts a filename and anoptional "options" stringfor specifyingthe header, footer, fonts, etc.

To make a "portal" script, addthe following codetocompletethe 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);
```

CallingHTMLDOCfrom C

C programs offerthe best flexibility and easily supports on-the-fly report generation withoutthe needfor temporaryfiles.

Here are some simple C functions that can beusedto generate aPDF reporttothe HTTP clientfrom atemporaryfileor 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"));
}
```

CallingHTMLDOCfrom Java

Java programs are a portable wayto addPDF supporttoyour web server. Here is a class called *htmldoc* that acts as a portal, convertingthe namedfileto PDF. It can also be called byyour Java servletsto process an HTMLfile and sendthe resulttothe client inPDF 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 ofthe command-line options supported by HTMLDOC.

Basic Usage

The basic command-line usageforHTMLDOC is:

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

The first form convertsthe namedHTMLor Markdown filestothe specified outputformat immediately. The second form loadsthe specified `.book`file and displaystheHTMLDOC window, allowing a userto make changesand/or generatethe document interactively.

If no outputfileordirectory is specified, then all output is senttothe standard output file.

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

Options

The following command-line options are recognized by HTMLDOC.

-ddirectory

The `-d` option specifies an outputdirectoryforthe documentfiles.

This option is not compatible withthe EPUBorPDF output formats.

-f filename

The -f option specifies an outputfileforthe document.

-tformat

The -t option specifiesthe outputformatforthe document and can be one ofthe following:

Format	Description
epub	Generate an EPUB file.
html	Generate oneor moreindexedHTMLfiles.
htmlsep	Generate separateHTML filesfor each heading inthe table-of-contents.
pdf	Generate aPDFfile (default version - 1.4).
pdf11	Generate aPDF 1.1filefor Acrobat Reader 2.0 and later.
pdf12	Generate aPDF 1.2filefor Acrobat Reader 3.0 and later.
pdf13	Generate aPDF 1.3filefor Acrobat Reader 4.0 and later.
pdf14	Generate aPDF 1.4filefor Acrobat Reader 5.0 and later.
ps	Generate oneor morePostScript files (default level - 2).
ps1	Generate oneor more Level 1PostScriptfiles.
ps2	Generate oneor more Level 2PostScriptfiles.
ps3	Generate oneor more Level 3PostScriptfiles.

-v

The -v option specifies that progress information should be sent/displayedtothe standard error file.

--batch filename.book

The --batch option specifies a bookfile that you would liketo generate withoutthe GUI popping up. This option can be combined with other optionsto generatethe 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 --bodycoloroption specifiesthe background colorfor all pages inthe document. The color can be specified by a standardHTML color nameor as a 6-digit hexadecimal number ofthe form #RRGGBB.

--bodyfont typeface

The `--bodyfont`option specifiesthe default text fontusedfor text inthe document body. The `typeface` parameter can be one ofthe 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 specifiesthe background imagefor all pages inthe document. The supported formats are GIF, JPEG, and PNG.

--book

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

--bottom margin

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

Thisoption is only available when generatingPostScriptorPDFfiles.

--browserwidth pixels

The `--browserwidth`option specifiesthebrowser width in pixels. Thebrowser width isusedto scale images and pixel measurements when generatingPostScript andPDFfiles. It does not affectthe font size of text.

The defaultbrowser width is 680 pixels which corresponds roughlyto a 96 DPI display. Please note thatyourimages and table sizes are equaltoor smaller thanthebrowser width,oryour output will overlapor truncate in places.

--charset charset

The `--charset`option specifiesthe 8-bit character set encodingto useforthe entire document.HTMLDOC comes withthe 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 limitedtothe first 128 Unicode characters found inthe input.

--color

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

Thisoption is only available when generatingPostScriptorPDFfiles.

--compression[=level]

The `--compression`option specifies that Flate compression should be performed onthe output file(s). Theoptional `level`parameter is a numberfrom 1 (fastest and least amount of compression)to 9 (slowest and most amount of compression).

Thisoption is only available when generatingPDFor Level 3PostScriptfiles.

--continuous

The `--continuous`option specifies thatthe input files comprise a web page (or site) and that notitle pageor table-of-contents should be generated. Unlikethe `--webpage`option described later in this chapter, page breaks are not inserted between each input file.

Thisoption is only available when generatingPostScriptorPDFfiles.

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

The `--cookies`option specifies oneor more HTTP cookies that should be sent when converting remote URLs. Each cookie must be separatedfromthe others by a semicolon and a space, and values containing whitespaceorthe semicolon must be placed inside double-quotes. When specifying multiple cookies,the entire cookie string must be surrounded by single quotes in orderforthe stringto be processed correctly.

--datadirdirectory

The `--datadir`option specifiesthe location of data filesused by HTMLDOC.

--duplex

The `--duplex`option specifies thatthe output should be formattedfor two sided printing.

Thisoption is only available when generatingPostScriptorPDFfiles. Usethe `--pscommands`optionto generatePostScript duplex mode commands.

--effectduration seconds

The `--effectduration`option specifiesthe duration of apage transition effect in seconds.

Thisoption is only available when generatingPDFfiles.

--embedfonts

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

--encryption

The `--encryption`option enables encryption and security featuresforPDF output.

Thisoption is only available when generatingPDFfiles.

--firstpagepage

The `--firstpage`option specifiesthe firstpage that will be displayed in aPDF file.The `page`parameter can be one ofthe following:

page	Description
p1	The firstpage ofthe document.
toc	The firstpage ofthe table-of-contents.
c1	The firstpage of chapter 1.

Thisoption is only available when generatingPDFfiles.

--fontsize size

The `--fontsize`option specifiesthe base font sizeforthe entire document in points (1 point = 1/72nd inch).

--fontspacing spacing

The `--fontspacing`option specifiestheline spacingforthe entire document as a multiplier ofthe base font size. A `spacing` value of 1 makes eachline of textthe same height asthe font.

--footer lcr

The `--footer`option specifiesthe contents ofthepage footer.The `lcr`parameter is a three-character string representingthe left, center, and right footer fields. Each charactercan be one ofthe following:

lcr	Description
.	A period indicates thatthe field should be blank.
:	A colon indicates thatthe field should containthe current and total number of pages inthe chapter (n/N).
/	A slash indicates thatthe field should containthe current and total number ofpages (n/N).
1	The number 1 indicates thatthe field should containthe currentpage number in decimalformat (1, 2, 3, ...)
a	A lowercase "a" indicates thatthe field should containthe currentpage numberusing lowercase letters.
A	An uppercase "A" indicates thatthe field should containthe currentpagenumberusing UPPERCASE letters.
c	A lowercase "c" indicates thatthe field should containthecurrent chapter title.
C	An uppercase "C" indicates thatthe field should containthecurrent chapterpage number.
d	A lowercase "d" indicates thatthe field should containthecurrent date.
D	An uppercase "D" indicates thatthe field should containthecurrent date and time.
h	An "h" indicates thatthe field should containthecurrent heading.
i	A lowercase "i" indicates thatthe field should containthecurrentpagenumber in lowercase roman numerals (i, ii, iii, ...)
I	An uppercase "I" indicates thatthe field should containthecurrentpagenumber in uppercase roman numerals (I, II, III, ...)
l	Alowercase "l" indicates thatthe field should containthe logo image.
l	An uppercase "L" indicates thatthe field should containthe logo image as a letterhead (shown at full size).
t	Alowercase "t" indicates thatthe field should containthe document title.
T	An uppercase "T" indicates thatthe field should containthecurrent time.
u	Alowercase "u" indicates thatthe field should containthecurrent filenameor URL.

Settingthe footerto "`...`" disablesthe footer entirely.

--formatformat

The --formatoption specifiesthe outputformatforthe document andcan be oneofthe following:

Format	Description
epub	Generate an EPUB file.
html	Generate oneor moreindexedHTMLfiles.
htmlsep	Generate separateHTML filesfor each headinginthe table-of-contents.
pdf	Generate aPDFfile (default version - 1.4).
pdf11	Generate aPDF 1.1filefor Acrobat Reader 2.0 and later.
pdf12	Generate aPDF 1.2filefor Acrobat Reader 3.0 and later.
pdf13	Generate aPDF 1.3filefor Acrobat Reader 4.0 and later.
pdf14	Generate aPDF 1.4filefor Acrobat Reader 5.0 and later.
ps	Generate oneor morePostScript files (default level - 2).
ps1	Generate oneor more Level 1PostScriptfiles.
ps2	Generate oneor more Level 2PostScriptfiles.
ps3	Generate oneor more Level 3PostScriptfiles.

--gray

The --grayoption specifies that grayscale output is desired.

Thisoption is only available when generatingPostScriptorPDFfiles.

--header lcr

The --headeroption specifiesthe contentsofthepage header.The lcrparameter is a three-characterstring representingthe left, center, and right header fields. Seethe --footeroptionforthe listof formatting characters.

Settingthe headerto "..." disablesthe header entirely.

--header1 lcr

The --header1option specifiesthe contentsofthepage headerforthe first body/chapter page.The lcrparameter is athree-characterstring representingthe left, center, and right header fields. Seethe --footeroptionforthelistof formatting characters.

Settingthe headerto "..." disablesthe firstpage 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 setsthe typeface that isusedforheadingsinthe document.The `typeface`parametercan be oneofthe 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 allofthe available optionstothe standard output file.

--helpdirdirectory

The `--helpdir`option specifiesthe locationofthe on-line helpfiles.

--hfimage*N* filename

The `--hfimage`*N*option specifies an imageto useinthe headerand/or footer, where N is anumberfrom 1to 10.The supported formats are GIF, JPEG, and PNG.

--jpeg[=quality]

The `--jpeg`option enables JPEG compressionof continuous-tone images.Theoptional `quality`parameter specifiesthe output qualityfrom 0 (worst)to 100 (best).

Thisoption is only available when generatingPDFor Level 2 and Level 3PostScriptfiles.

--landscape

The `--landscape`option specifies thatthe output should bein landscape orientation (long edge on top).

Thisoption is only available when generatingPostScriptorPDFfiles.

--left margin

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

Thisoption is only available when generatingPostScriptorPDFfiles.

--letterhead filename

The --letterheadoption specifiestheletterhead imageforthepage headers and footersforPostScript andPDFfiles.The supported formats are GIF, JPEG,and PNG.

> **Note:** You needto usethe --headerand/or --footer options withthe Lparameteror usethe correspondingHTMLpage commentsto displaythe logo imageinthe headeror footer.

--linkcolor color

The --linkcoloroption specifiesthe colorof linksin EPUB, HTML.andPDF output.The colorcan bespecified by nameor as a 6-digit hexadecimalnumberofthe form #RRGGBB.

--links

The --linksoption specifies thatPDF output should contain hyperlinks.

--linkstyle style

The --linkstyleoption specifiesthe styleof linksin EPUB, HTML,andPDF output.The stylecanbe "plain"for no decorationor "underline"to underline links.

--logoimage filename

The --logoimageoption specifiesthe logo imagefortheHTML navigation barandpage headersand footersforPostScriptandPDFfiles.The supported formats are GIF, JPEG,and PNG.

> **Note:** You needtousethe --headerand/or --footer options withthe lparameterorusethe correspondingHTMLpage commentsto displaythe logo imageinthe headeror footer.

--no-compression

The --no-compressionoption specifies that Flate compression should notbe performed onthe outputfiles.

--no-duplex

The --no-duplexoption specifies thatthe output shouldbe formattedfor one sided printing.

Thisoption is only available when generatingPostScriptorPDFfiles. Usethe --pscommandsoptionto generatePostScript duplex mode commands.

--no-embedfonts

The --no-embedfontsoption specifies that fonts should notbe embeddedinPostScriptandPDF output.

--no-encryption

The --no-encryptionoption specifies that no encryption/security features shouldbe enabledinPDF output.

Thisoption is only available when generatingPDFfiles.

--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.

--outdirdirectory

The `--outdir`option specifies anoutputdirectoryforthe documentfiles.

Thisoption is not compatible withthePDFoutput format.

--outfile filename

The `--outfile`option specifies anoutputfileforthe document.

--owner-password password

The `--owner-password`option specifiesthe owner passwordfor aPDF file. If notspecifiedortheemptystring (""), a random password is generated.

Thisoption is only availablewhen generatingPDFfiles.

--pageduration seconds

The `--pageduration`option specifiesthenumberof seconds that eachpage willbe displayedinthe document.

Thisoption is only availablewhen generatingPDFfiles.

--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 specifiesthe document permissions.The available permission parameters are listed below:

Permission	Description
all	All permissions
annotate	Usercan annotate document
copy	Usercan copy textandimagesfrom document
modify	Usercan modify document
print	Usercan print document
no-annotate	User cannot annotate document
no-copy	User cannot copy textandimagesfrom document
no-modify	User cannot modify document
no-print	User cannot print document
none	No permissions

The `--encryption`option mustbeusedin conjunction withthe `--permissions` parameter.

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

Multiple optionscanbespecified by separating them with commas:

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

Thisoption is only availablewhen generatingPDFfiles.

--portrait

The `--portrait`option specifies thattheoutput shouldbein portrait orientation (short edge on top).

Thisoption is only availablewhen generatingPostScriptorPDFfiles.

--pdf-indent margin

The `--pdf-indent`option specifiesthe indentationfor pre-formatted content.The default units are points (1 point = 1/72nd inch);thesuffixes "in", "cm",and "mm" specify inches, centimeters,and millimeters, respectively.

--pscommands

The `--pscommands`option specifies thatPostScript device commands shouldbe writtentotheoutputfiles.

Thisoption is only availablewhen generating Level 2and Level 3PostScriptfiles.

--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 setsthe typeface that isusedfor textinthe document.The `typeface`parametercanbeoneofthe 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 atitlepage shouldbe generated.

--titlefile filename

The `--titlefile`option specifies aHTMLor Markdownfiletouseforthetitle page.

--titleimage filename

The `--titleimage`option specifiesthetitle imageforthetitle page.The supported formats are GIF, JPEG,and PNG.

--tocfooter lcr

The `--tocfooter`option specifiesthe contentsofthe table-of-contents footer.The `lcr`parameter is athree-characterstring representingthe left, center,and right footer fields. Seethe `--footer`optionforthelistof formatting characters.

Settingthe TOC footerto "`...`" disablesthe TOC footer entirely.

--tocheader lcr

The `--tocheader`option specifiesthe contentsofthe table-of-contents header.The `lcr`parameter is athree-characterstring representingthe left, center,and rightheader fields. Seethe `--footer`optionforthelistof formatting characters.

Settingthe TOCheaderto "`...`" disablesthe TOCheader entirely.

--toclevels levels

The `--toclevels` options specifiesthenumberof heading levelsto includeinthe table-of-contents pages.The `levels`parameter is anumberfrom1to 6.

--toctitlestring

The `--toctitle` options specifiesthestringto display atthe topofthe table-of-contents;the defaultstring is "Tableof Contents".

--top margin

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

Thisoptionis only availablewhen generatingPostScriptorPDFfiles.

--user-password password

The `--user-password`option specifiesthe user passwordfor aPDF file. If notspecifiedortheemptystring (""), no password willbe requiredto viewthe document.

Thisoptionis only availablewhen generatingPDFfiles.

--verbose

The `--verbose`option specifies that progress information shouldbe sent/displayedtothe standard error file.

--version

The `--version`option displaystheHTMLDOC version number.

--webpage

The `--webpage`option specifies thatthe input files comprise a webpage (or site)and that notitlepageor table-of-contents shouldbe generated.HTMLDOC will insert apage break between each input file.

Thisoptionis only availablewhen generatingPostScriptorPDFfiles.

--xrxcomments

The `--xrxcomments`option specifies that XeroxPostScript job comments shouldbe writtentotheoutputfiles.

Thisoptionis only availablewhen generatingPostScriptfiles.

Environment Variables

HTMLDOC looksfor several environment variables whichcan overridethe default directories, display additional debugging information,and disable CGI mode.

HTMLDOC_DATA

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

HTMLDOC_DEBUG

This environment variable enables debugging information thatis sentto stderr.The valueis alistof keywords separated by spaces:

keyword	Information Shown
`links`	Shows allofthe linksin a document
`memory`	Shows memory usage statistics
`remotebytes`	Showsthenumberof bytes that were transferred via HTTP
`table`	Puts a boxaround each table, row,and cell
`tempfiles`	Showsthetemporary files that were created,and preserves themfor debugging
`timing`	Showsthe loadand render times
`all`	Allofthe above

HTMLDOC_HELP

This environment variable specifiesthe locationof 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. Itis most usefulforusingHTMLDOC on a web serverfrom a scripting languageor invocationfrom a program.

Messages

HTMLDOC sends errorand status messagesto stderr unlessthe `--quiet`optionis provided onthe command-line. Applicationscan capturethese messagesto relay errorsor statisticstothe user.

BYTES: Message

The `BYTES:` message specifiesthenumberof bytes that werewrittento anoutput file. Iftheoutputis directed at adirectory then multiple `BYTES:` messages willbe sent.

DEBUG: Messages

The `DEBUG:` messages contain debugging information based onthe valueofthe `HTMLDOC_DEBUG` environment variable. Normally, no `DEBUG:` messages are sent by HTMLDOC.

ERRnnn: Messages

The `ERRnnn:` messages specify an error condition. Error numbers1to 14 maptothe following errors:

1. No files were foundor loadable.
2. Nopages were generated.
3. The document contains too many filesor chapters.
4. HTMLDOC ran outof memory.
5. Thespecifiedfile couldnotbe found.
6. The comment contains a badHTMLDOC formatting command.
7. The imagefileisnotin a known format.
8. HTMLDOCwas unableto remove atemporary file.
9. HTMLDOC had an unspecified internal error.
10. HTMLDOC encountered a networking errorwhen retrieving afile via aURL.
11. HTMLDOCwas unableto read a file.
12. HTMLDOCwas unableto write a file.
13. AHTML errorwas foundin asource file.
14. A table,image,or text fragmentwas too largeto fitinthe space provided.
15. A hyperlinkinthesource fileswas unresolved.
16. A header/footerstringinthe document contains a bad $ command.

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

INFO: Messages

The `INFO:` messages contain general information thatis loggedwhenHTMLDOCisrunningin CGI modeorwhen youusethe `--verbose` option.

PAGES: Message

The `PAGES:` message specifiesthenumberofpages that werewrittento anoutput file. Iftheoutputisdirected at adirectory then multiple `PAGES:` messages willbe sent. No `PAGES:` messages are sentwhen generatingHTMLor EPUBoutput.

REMOTEBYTES: Message

The `REMOTEBYTES:` message specifiesthenumberof bytes that were transferredusing HTTP. This messageis only displayed ifthe `HTMLDOC_DEBUG` environment variable hasthe 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 alloftheHTML elementsand attributes that are recognizedand supported by HTMLDOC.

General Usage

There are two typesofHTML files - structured documentsusingheadings (H1, H2, etc.) whichHTMLDOC calls "books",and unstructured documents that donotuseheadings whichHTMLDOCcalls "web pages".

A verycommon mistakeisto try converting a webpage using:

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

which will likely produce aPDFfile with nopages. Toconvert webpage files you **must**usethe `--webpage`option atthe command-lineor choose *Web Page*inthe input tabofthe GUI.

Note:HTMLDOC doesnot supportHTML 4.0 elements, attributes, stylesheets,or scripting.

Elements

The followingHTML elements are recognized by HTMLDOC:

Element	Version	Supported?	Notes
!DOCTYPE	3.0	Yes	DTDis 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 specialHTML commentsto initiatepage breaks, settheheaderand footertext,and controlthecurrent media options:

`<!-- FOOTER LEFT "foo" -->`
Setsthe left footer text;the testis appliedtothecurrentpage if empty,orthe nextpage otherwise.
`<!-- FOOTER CENTER "foo" -->`
Setsthe center footer text;the testis appliedtothecurrentpage if empty,orthe nextpage otherwise.
`<!-- FOOTER RIGHT "foo" -->`
Setsthe right footer text;the testis appliedtothecurrentpage if empty,orthe nextpage otherwise.
`<!-- HALF PAGE -->`
Breaktothe next half page.
`<!-- HEADER LEFT "foo" -->`
Setsthe leftheader text;the testis appliedtothecurrentpage if empty,orthe nextpage otherwise.
`<!-- HEADER CENTER "foo" -->`
Setsthe centerheader text;the testis appliedtothecurrentpage if empty,orthe nextpage otherwise.

`<!-- HEADER RIGHT "foo" -->`
Setsthe rightheader text;the testis appliedtothecurrentpage if empty,orthe nextpage otherwise.

`<!-- MEDIA BOTTOM nnn -->`
Setsthe bottom marginofthe page.The "nnn"stringcanbe any standard measurement value, e.g. 0.5in, 36, 12mm, etc. Breaksto a newpage ifthecurrentpageis already marked.

`<!-- MEDIA COLOR "foo" -->`
Setsthe media color attributeforthe page.The "foo"stringis any color name thatis supported bythe printer, e.g. "Blue", "White", etc. Breaksto a newpageor sheet ifthecurrentpageis already marked.

`<!-- MEDIA DUPLEX NO -->`
Chooses single-sided printingforthe page; breaksto a newpageor sheet ifthecurrentpageis already marked.

`<!-- MEDIA DUPLEX YES -->`
Chooses double-sided printingforthe page; breaksto a new sheet ifthecurrentpageis alreadymarked.

`<!-- MEDIA LANDSCAPE NO -->`
Chooses portrait orientationforthe page; breaksto a newpage ifthecurrentpageisalreadymarked.

`<!-- MEDIA LANDSCAPE YES -->`
Chooses landscape orientationforthe page; breaksto a newpage ifthecurrentpageisalreadymarked.

`<!-- MEDIA LEFT nnn -->`
Setsthe left marginofthe page.The "nnn"stringcanbe any standard measurement value, e.g.0.5in, 36, 12mm, etc. Breaksto a newpageifthecurrentpageisalreadymarked.

`<!-- MEDIA POSITION nnn -->`
Setsthe media position attribute (input tray)forthe page.The "nnn"stringis an integer that usually specifiesthe tray number. Breaksto a newpageor sheetifthecurrentpageisalreadymarked.

`<!-- MEDIA RIGHT nnn -->`
Setsthe right marginofthe page.The "nnn"stringcanbe any standard measurement value, e.g.0.5in,36, 12mm, etc. Breaksto a newpageifthecurrentpageisalreadymarked.

`<!-- MEDIA SIZE foo -->`
Setsthe media sizetothespecifiedsize.The "foo"stringcanbe "Letter", "Legal", "Universal",or "A4"for standard sizesor "WIDTHxHEIGHTunits"for custom sizes,e.g. "8.5x11in"; breaksto a newpageor sheetifthecurrentpageisalreadymarked.

`<!-- MEDIA TOP nnn -->`
Setsthe top marginofthe page.The "nnn"stringcanbe any standard measurement value,e.g.0.5in,36, 12mm, etc. Breaksto a newpageifthecurrentpageisalreadymarked.

`<!-- MEDIA TYPE "foo" -->`
Setsthe media type attributeforthe page.The "foo"stringis any type name thatis supported bytheprinter,e.g. "Plain", "Glossy", etc. Breaksto a newpageor sheetifthecurrentpageisalreadymarked.

`<!-- NEED length -->`
Breakif thereis less than `length` units left onthecurrent page.The `length` value defaultsto linesof text butcanbe suffixedby `in`, `mm`,or `cm`toconvertfromthe corresponding units.

`<!-- NEW PAGE -->`
Breaktothe next page.

`<!-- NEW SHEET -->`
Breaktothe next sheet.

`<!-- NUMBER-UP nn -->`
Setsthenumberofpages that are placed on eachoutput page. Valid values are 1, 2, 4, 6, 9,and 16.

`<!-- PAGE BREAK -->`
Breaktothe next page.

Header/Footer Strings

The HEADERand FOOTER comments allow youto set an arbitrarystringof textforthe left, center,and right headersand footers. Eachstring consistsofplain text; special valuesor stringscanbe insertedusingthe dollar sign ($):

$$
: Inserts a single dollarsigninthe header.

$CHAPTER
: Insertsthecurrent chapter heading.

$CHAPTERPAGE
$CHAPTERPAGE(format)
: Insertsthecurrentpagenumber within a chapteror file. When aformatis specified, uses that numericformat(1 = decimal, i =lowercase roman numerals, I = uppercase roman numerals, a =lowercase ascii, A= uppercase ascii)forthepage numbers.

$CHAPTERPAGES
$CHAPTERPAGES(format)
: Insertsthe totalpage count within a chapteror file. When aformatis specified, uses thatnumericformat(1 = decimal, i =lowercase roman numerals, I = uppercase roman numerals, a =lowercase ascii, A= uppercase ascii)forthepage count.

$DATE
$DATE(format)
: Insertsthecurrent date. See Date/Time Formatsfor details ontheformat string. When noformatis supplied,the default dateformatforthecurrent localeis used.

$HEADING
: Insertsthecurrent heading.

$HFIMAGE1
$HFIMAGE2
$HFIMAGE3
$HFIMAGE4
$HFIMAGE5
$HFIMAGE6
$HFIMAGE7
$HFIMAGE8
$HFIMAGE9
$HFIMAGE10
: Insertsthespecified header/footer image; all other textinthestring willbeignored.

$LETTERHEAD
: Insertsthe logo image as aletterhead with no down-scaling; all other textinthestring willbeignored.

$LOGOIMAGE
: Insertsthe logo image; all other textinthestring willbeignored.

$PAGE
$PAGE(format)
: Insertsthecurrentpage number. When aformatis specified, uses thatnumericformat(1 = decimal, i =lowercase roman numerals, I= uppercase roman numerals, a=lowercase ascii, A= uppercase ascii)forthepage numbers.

$PAGES
$PAGES(format)
: Insertsthe totalpage count. When aformatis specified, uses thatnumericformat(1 = decimal, i =lowercase roman numerals, I= uppercase roman numerals, a=lowercase ascii, A= uppercase ascii)forthepage count.

`$TIME`
`$TIME(format)`
: Insertsthecurrent time. See Date/Time Formatsfor details ontheformat string. When noformatis supplied,the default timeformatforthecurrent localeis used.

`$TITLE`
: Insertsthe document title.

`$URL`
: Insertsthe document filenameorURL.

Date/Time Formats

The `$DATE`and `$TIME` header/footer strings support anoptionalformatstringin parenthesis. Letters represent date/time values while other characters are inserted verbatim.The following letters are supported:

Letter	Description
A	Full weekday name
a	Abbreviated weekday name
B	Full month name
b	Abbreviated month name
C	Century (CC)
c	Default dateand timeformat
d	Dayofthe month ("01"to "31")
e	Dayofthe month (" 1"to "31")
F	YYYY-MM-DD
H	Hoursfor 24-hour clock ("00"to "23")
I	Hoursfor 12-hour clock ("01"to "12")
j	Dayofthe year ("001"to "366")
k	Hoursfor 24-hour clock (" 0"to "23")
l	Hoursfor 12-hour clock (" 1"to "12")
M	Minutes ("00"to "59")
m	Monthnumber ("01"to "12")
p	"am"or "pm"
R	Hoursand minutes ("HH:MM")
r	Hours, minutes, seconds,and am/pm ("HH:MM:SS am/pm")
S	Seconds ("00"to "60")
T	Hours, minutes,and seconds ("HH:MM:SS")
X	Default timeformat
x	Default dateformat

Letter	Description
Y	Year with century (CCYY)
y	Year without century (YY)
Z	Time zone name
z	Time zone offsetfrom UTC

FONT Attributes

Limited typeface specificationis currently supportedto ensure portability across platformsandfor olderPostScript 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 aresilentlyignored.

Headings

CurrentlyHTMLDOC supports a maximumof 1000 chapters (H1 headings).This limitcanbe increasedby changingthe MAX_CHAPTERS constantinthe *config.h*file included withthesource code.

All chapters start with a top-level heading (H1) markup. Anyheadings within a chapter mustbeof a lower level (H2to H15). Each chapter starts a newpageorthe next odd-numberedpageif duplexingis selected.

> **Note:** Heading levels 7to 15 arenot standardHTMLand willnot likelybe recognizedby most web browsers.

Theheadings youuse within a chapter must start atlevel 2 (H2). If you skip levelsthe heading willbe shown underthe lastlevel thatwas known. For example,if youusethe following hierarchyof 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>
...
```

thetable-of-contents thatis generated will show:

- **Chapter Heading**
  - ♦ Section Heading1
  - ♦ Section Heading 2
    - ◊ Sub-Section Heading1
      - · Sub-Sub-Section Heading1
      - · Sub-Sub-Section Heading 2
    - ◊ Sub-Section Heading 2
      - · Sub-Sub-Section Heading 3
  - ♦ Section Heading 3

Numbered Headings

Whenthe numberedheadingsoptionis enabled,HTMLDOC recognizesthe followingadditional attributesfor all heading elements:

`VALUE="#"`
: Specifiesthe starting valuefor this headinglevel (defaultis "1"forall new levels).

`TYPE="1"`
: Specifies thatdecimal numbers shouldbe generatedfor this heading level.

`TYPE="a"`
: Specifies thatlowercase letters shouldbe generatedfor this heading level.

`TYPE="A"`
: Specifies that uppercase letters shouldbe generatedfor this heading level.

`TYPE="i"`
: Specifies thatlowercase roman numerals shouldbe generatedfor this heading level.

`TYPE="I"`
: Specifies that uppercase roman numerals shouldbe generatedfor this heading level.

Images

HTMLDOC supports loadingof GIF, JPEG,and PNG imagefiles. BMP image supportis deprecatedand willbe removedin a future versionof HTMLDOC. EPSand other typesof image files arenot supported at this time.

Links

External URLand internal (#targetand filename.html) links are fully supportedforHTMLandPDFoutput.

When generatingPDF files, localPDFfile linkswillbe convertedto externalfile linksforthePDF viewer insteadof URL links. That is, youcan directly linkto another localPDFfilefromyourHTML document with:

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

META Attributes

HTMLDOC supportsthe following META attributesforthetitlepageand document information:

<META NAME="AUTHOR" CONTENT="..."
: Specifiesthe document author.

<META NAME="COPYRIGHT" CONTENT="..."
: Specifiesthe document copyright.

<META NAME="DOCNUMBER" CONTENT="..."
: Specifiesthe document number.

<META NAME="GENERATOR" CONTENT="..."
: Specifiesthe application that generatedtheHTML file.

<META NAME="HTMLDOC.filename" CONTENT="..."
: Specifiesthe filename thatis reportedin CGI mode.

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

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

Tables

CurrentlyHTMLDOC supports a maximumof 200 columns within a single table.This limitcanbe increasedby changingthe MAX_COLUMNS constantinthe *config.h*file included withthesource code.

HTMLDOC doesnot supportHTML 4.0 table elementsor attributes, such as TBODY, THEAD, TFOOT,or RULES.

Chapter 5 - Markdown Reference

This chapter describesthe markdown syntax thatis recognizedand supportedby HTMLDOC.

General Syntax

Markdownis a simple plain-textformat that uses formatting conventions that are commonlyusedin emailand other text-based communications. Markdownisusedbymostofthe major blogging, web site,and project hosting platformsandis supportedby many standalone text editors.

HTMLDOC supportsthe CommonMark versionof markdown syntax withthe following exceptions:

- Metadata asusedby Jekylland other web markdown solutionscanbe placed atthe beginningofthe file;
- "@" linkscanbeused which resolvetoheadings withinthe file;
- Tablescanbe embeddedusingthe "|" separator;
- EmbeddedHTML markupand entities are explicitlynot supportedor allowed;
- Tabs aresilently expandedtothe markdown standardof four spaces sinceHTML uses eight spaces per tab;and
- Some pathological nested linkand inline style features supportedby CommonMark (`******Really Strong Text******`) arenot supportedby `mmd`.

  Note:HTMLDOC doesnot support embeddedHTMLin markdown documents becausethe versionofHTML (or XHTML) cannotbe reliably determined, making supportof certain character entitiesand language elements problematic.

Metadata Syntax

Metadataisspecified atthe topof a markdownfile 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 supportsthe "author", "copyright", "language", "subject", "title",and "version" metadataandsilently ignores everything else.

Link Targetsand @ Links

CommonMark defines no standardfor how implementationsgenerate anchorsor identifiersforheadingsin a markdownfile -this makes hyperlinkingto a named section within a document basically impossible. Jekylland othermarkdown implementations allowthe special link "@"tobe used, whichHTMLDOC supports:

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

...

# Screwing in a Light Bulb

...
```

To reference a markdown headingfrom aHTML file,convertthe headingto lowercase, replace spaces withthe hyphen ("-"),and remove any special characters. Thus, aHTMLfile would referencethe previous headingusingthe following HTML:

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

Table Syntax

CommonMark doesnot define a syntaxfor plain-text tables,instead relying on embeddedHTML whichHTMLDOC doesnot support. Both GithubandJekyll support acommon markdown extensionforplain text tables that usesthe vertical pipe ("|") characterto specify column separations.The firstline containsthe table header,the secondlineis a horizontal separator,andthe remaining lines containthe 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:

Heading1	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 pipescanbe 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 tableheadings are always centered, youcan controlthe alignmentofthe body cellsbyusingthe colon (":") characterinthe separator line. Put a leading colonto specify left alignment (the default), a trailing colonfor right alignment,or bothto 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 donot needtobe padded so that theyline up -the following (less readable) exampleis 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 licensesformost software are designedto take awayyour freedomto shareand change it. By contrast,the GNU General Public Licenseis intendedto guaranteeyour freedomto shareand change free software--to make surethe softwareis freeforall its users.This General Public License appliestomostofthe Free Software Foundation's softwareandto any other program whose authors committousing it. (Some other Free Software Foundation softwareis coveredbythe GNU Library General Public License instead.) Youcan apply ittoyour programs, too.

When we speakof free software, we are referringto freedom,not price. Our General Public Licenses are designedto make sure that you havethe freedomto distribute copiesof free software (and chargeforthis serviceif you wish), that you receivesource codeorcan get itif you want it, that youcan changethe softwareoruse piecesof itin new free programs;and that you know youcan dothese things.

To protectyour rights, we needto makerestrictions that forbid anyoneto deny youthese rightsorto ask youto surrenderthe rights. Theserestrictions translateto certain responsibilitiesforyouifyou distribute copiesofthe software,orifyou modify it.

For example,if you distributecopiesof such a program, whether gratisorfor a fee, you must givethe recipientsallthe rights that you have. You must make sure that they, too, receiveorcan getthesource code. And you must show themtheseterms so they know their rights.

We protectyour rightswith two steps: (1) copyrightthe software,and (2) offer youthis license which gives you legal permissionto copy, distributeand/or modifythe software.

Also,for each author's protectionand ours, we wantto make certain that everyone understands that thereis no warrantyforthis free software. Ifthe softwareis modifiedby someone elseand passed on, we want itsrecipientsto know that what they haveisnotthe original, so that any problems introducedby otherswillnot reflect onthe original authors' reputations.

Finally, any free programis threatened constantlyby software patents. We wishto avoidthe danger that redistributorsof a free programwill individually obtain patent licenses,in effect makingthe program proprietary. To prevent this, we have made it clear that any patent mustbe licensedfor everyone's freeuseornot licensed at all.

The precisetermsand conditionsfor copying, distributionand modification follow.

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

0. This License appliesto any programor other work which contains a notice placedbythe copyright holder saying it maybe distributedunderthetermsofthis General Public License.The "Program", below, refersto any such programor work,and a "work based onthe Program" means eitherthe Programor any derivative workunder copyright law: thatisto say, a work containingtheProgramor a portionof it, either verbatimorwith modificationsand/or translated into another language. (Hereinafter, translationis included without limitationinthe term "modification".) Each licenseeis addressed as "you".

   Activities other than copying, distributionand modification arenot coveredbythis License; they are outside its scope.The actofrunningtheProgramisnot restricted,andtheoutputfromtheProgramis covered onlyif its contents constitute a work based ontheProgram (independentof having been madebyrunningthe Program). Whether thatis true depends on whattheProgram does.
1. You may copyand distribute verbatimcopiesofthe Program'ssource code asyou receive it,in any medium, provided thatyou conspicuouslyand appropriately publish on each copy an appropriate copyright noticeand disclaimerof warranty; keep intactallthe notices that refertothis Licenseandtothe absenceof any warranty;and give any otherrecipientsoftheProgram a copyofthisLicense alongwiththe Program.

   You may charge a feeforthe physical actof transferring a copy,andyou may atyouroption offerwarranty protectionin exchangefor a fee.
2. You may modifyyourcopyorcopiesoftheProgramor any portionof it, thus forming a work based onthe Program,andcopyand distribute such modificationsor workunderthetermsof Section1 above, provided thatyou also meetallofthese conditions:
   a. You must causethe modified filesto carry prominent notices stating thatyou changedthe filesandthe dateof any change.
   b. You must cause any work thatyou distributeor publish, thatin wholeorin part containsoris derivedfromtheProgramor any part thereof,tobe licensed as a whole at no chargetoall third partiesunderthetermsofthis License.
   c. ifthe modified program normally reads commands interactivelywhen run,you must cause it,when startedrunningfor such interactiveuseinthemost ordinary way,to printor display an announcement includingan appropriatecopyright noticeand a notice that thereis nowarranty (or else, saying thatyou provide a warranty)and that users may redistributethe programunderthese conditions,and tellingthe user howto view acopyofthis License.

(Exception:iftheProgram itselfis interactive but doesnot normally print such an announcement,your work basedontheProgramisnot requiredto print an announcement.)

These requirementsapplytothe modified work as a whole. If identifiable sectionsof that work arenotderivedfromthe Program,andcanbe reasonably considered independentand separate worksin themselves, thenthis License,and its terms, donotapplyto those sectionswhenyou distribute them as separate works. Butwhenyoudistributethe same sections as partof a whole whichis a work basedonthe Program,the distributionofthe whole mustbeonthetermsofthis License, whose permissionsfor other licensees extendtothe entire whole,and thusto eachand every part regardlessof who wrote it.

Thus, itisnotthe intentofthis sectionto claim rightsor contestyour rightsto workwritten entirelyby you; rather,the intentisto exercisethe rightto controlthe distributionof derivativeor collective works basedonthe Program.

In addition, mere aggregationof another worknot basedontheProgramwiththeProgram (orwith a work basedonthe Program)on a volumeof a storageor distribution medium doesnot bringthe other workunderthe scopeofthis License.

3. You maycopyanddistributetheProgram (or a work basedon it,under Section 2)in object codeor executable formunderthetermsof Sections1and 2 above provided thatyou also dooneofthe following:
    a. Accompany itwiththecomplete corresponding machine-readablesource code, which mustbe distributedunderthetermsof Sections1and 2 aboveon a medium customarilyusedfor software interchange; or,
    b. Accompany itwith awritten offer, validfor at least three years,to give anythird party,for a chargeno more thanyour costof physically performingsource distribution, acomplete machine-readablecopyofthe correspondingsource code,tobe distributedunderthetermsof Sections1and 2 aboveon a medium customarilyusedforsoftware interchange; or,
    c. Accompany itwiththe informationyou received astothe offertodistribute correspondingsource code. (This alternativeis allowed onlyfor noncommercial distributionand onlyifyou receivedthe programin object codeor executable formwith such an offer,in accordwith Subsection b above.)

   Thesource codefor a work meansthe preferred formofthe workfor making modificationsto it. Foran executable work,completesource code meansallthesource codeforall modules it contains, plus any associated interface definition files, plusthe scriptsusedto control compilationand installationofthe executable. However, as a special exception,thesource code distributed neednot include anything thatis normally distributed (in eithersourceor binary form)withthe major components (compiler, kernel,and so on)ofthe operating systemon whichthe executable runs, unless that component itself accompaniesthe executable.

   Ifdistributionof executableor object codeis madeby offering accesstocopyfrom a designated place, then offering equivalent accesstocopythesource codefromthe sameplace counts asdistributionofthesource code, even thoughthird partiesarenot compelledtocopythesource alongwiththe object code.
4. You maynot copy, modify, sublicense,ordistributetheProgram except as expressly providedunderthis License. Any attemptotherwiseto copy, modify, sublicenseordistributetheProgramis void,andwill automatically terminateyour rightsunderthis License. However, parties who have received copies,or rights,fromyouunderthisLicensewillnot have their licenses terminated so long as such parties remainin full compliance.
5. Youarenot requiredto acceptthis License, sinceyouhavenot signed it. However, nothing else grantsyou permissionto modifyordistributetheProgramor its derivative works. These actionsare prohibitedby lawifyou donot acceptthis License. Therefore,by modifyingor distributingtheProgram (or any work basedonthe Program),you indicateyour acceptanceofthisLicenseto do so,andall itstermsand conditionsfor copying, distributingor modifyingtheProgramor works basedon it.

6. Each timeyou redistributetheProgram (or any work basedonthe Program),the recipient automatically receives a licensefromthe original licensorto copy,distributeor modifytheProgram subjecttothesetermsand conditions. You maynot impose any furtherrestrictionsonthe recipients' exerciseofthe rights granted herein. Youarenot responsiblefor enforcing compliancebythird partiestothis License.
7. If, as a consequenceof a court judgmentor allegationof patent infringementorfor any other reason (not limitedto patent issues), conditionsare imposedonyou (whetherby court order, agreementor otherwise) that contradictthe conditionsofthis License, they donot excuseyoufromthe conditionsofthis License. Ifyou cannotdistribute so asto satisfy simultaneouslyyour obligationsunderthisLicenseand any other pertinent obligations, then as a consequenceyou maynotdistributetheProgram at all.For example,if a patent license wouldnot permit royalty-free redistributionoftheProgrambyall those who receivecopies directlyor indirectly through you, thenthe only wayyou could satisfy both itandthisLicense wouldbeto refrain entirelyfromdistributionofthe Program.

   If any portionofthis sectionis held invalidor unenforceableunder any particular circumstance,the balanceofthe sectionis intendedtoapplyandthe section as a wholeis intendedtoapplyin other circumstances.

   Itisnotthe purposeofthis sectionto induceyouto infringe any patentsor other property right claimsorto contest validityof any such claims;this section hasthe sole purposeof protectingthe integrityofthe freesoftwaredistribution system, whichis implementedby public license practices. Many peoplehave made generous contributionstothe wide rangeofsoftware distributed through that systemin relianceon consistent applicationof that system; itis uptothe author/donorto decideif heor sheis willingtodistributesoftware through any other systemand a licensee cannot impose that choice.

   This sectionis intendedto make thoroughly clear whatis believedtobe a consequenceofthe restofthis License.
8. IfthedistributionandoruseoftheProgramis restrictedin certain countries eitherby patentsorby copyrighted interfaces,the originalcopyrightholder who placestheProgramunderthisLicense may addan explicit geographicaldistribution limitation excluding those countries, so thatdistributionis permitted onlyinor among countriesnot thus excluded. In such case,thisLicense incorporatesthe limitation asifwritteninthe bodyofthis License.
9. TheFree Software Foundation may publish revisedand/or newversionsofthe General PublicLicensefrom timeto time. Such newversionswillbe similarin spirittothe present version, but may differin detailto address new problemsor concerns.

   Eachversionis given a distinguishingversion number. IftheProgram specifies aversionnumberofthisLicense which appliesto itand "any later version",youhavetheoptionof followingthetermsand conditions eitherof thatversionorof any laterversion publishedbytheFree Software Foundation. IftheProgram doesnot specify aversionnumberofthis License,you may choose anyversion ever publishedbytheFree Software Foundation.
10. Ifyou wishto incorporate partsoftheProgram into other free programs whosedistribution conditionsare different, writetothe authorto askfor permission.Forsoftware whichiscopyrightedbytheFree Software Foundation, writetotheFree Software Foundation; we sometimes make exceptionsfor this. Our decisionwillbe guidedbythe two goalsof preservingthe free statusofall derivativesof ourfreesoftwareandof promotingthe sharingand reuseofsoftware 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 TOTHE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUMETHE 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 THIRDPARTIES OR A FAILURE OF THE PROGRAM TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OFTHE POSSIBILITY OF SUCH DAMAGES.

END OFTERMS AND CONDITIONS

Howto Apply These Termsto Your New Programs

Ifyou develop a new program,andyou want ittobeofthe greatest possibleusetothe public,the best wayto achievethisisto make it freesoftwarewhich everyonecan redistributeand changeunderthese terms.

To do so, attachthe following noticestothe program. Itis safestto attach themtothe startof eachsourcefiletomost effectively conveythe exclusionof warranty;andeachfile shouldhave at leastthe "copyright"lineand a pointerto wherethe full noticeis 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 informationon howto contactyouby electronicand paper mail.

Ifthe programis interactive, make itoutput a short notice likethiswhen it startsinan 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 showthe appropriate partsofthe General Public License. Of course,the commandsyouuse maybe called something other than `` `show w' `` and `` `show c' ``; they could evenbe mouse-clicksor menu items--whatever suitsyour program.

You should also getyour employer (ifyou work as a programmer)oryour school,if any,tosign a "copyright disclaimer"forthe program,if necessary. Hereis a sample; alterthe 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 describestheHTMLDOC *.book*file format.

Introduction

TheHTMLDOC *.book*fileformatis a simple textformat that providesthe command-line optionsand files thatare partofthe document. These filescanbeusedfromthe GUI interfaceorfromthe command-lineusingthe `--batch` option:

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

The first formwill loadthe bookand displaythe GUI interface,if configured. Windows users shoulduse *ghtmldoc.exe* executableto showthe GUIand *htmldoc.exe*forthe batch mode:

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

The Header

Each *.book*file startswith aline reading:

```
#HTMLDOC 1.9
```

Theversionnumber (1.9)is optional.

The Options

Followingtheheaderis aline containingthe optionsforthe book. Youcanuse any valid command-lineoptiononthis line:

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

Longoption linescanbe brokenusing a trailing backslash (\)onthe endof each continuation line:

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

The Files

Followingthe optionsare alistof filesorURLsto includeinthe 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 followingisthecomplete bookfile neededtogeneratethis 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
```