

# wkhtmltopdf 0.10.0 rc2 Manual

This file documents wkhtmltopdf, a program capable of converting html documents into PDF documents.

## Contact

If you experience bugs or want to request new features please visit <http://code.google.com/p/wkhtmltopdf/issues/list>, if you have any problems or comments please feel free to contact me: see <http://www.madalgo.au.dk/~jakobt/#about>

## Reduced Functionality

Some versions of wkhtmltopdf are compiled against a version of QT without the wkhtmltopdf patches. These versions are missing some features, you can find out if your version of wkhtmltopdf is one of these by running `wkhtmltopdf --version` if your version is against an unpatched QT, you can use the static version to get all functionality.

Currently the list of features only supported with patch QT includes:

- Printing more then one HTML document into a PDF file.
- Running without an X11 server.
- Adding a document outline to the PDF file.
- Adding headers and footers to the PDF file.
- Generating a table of contents.
- Adding links in the generated PDF file.
- Printing using the screen media-type.
- Disabling the smart shrink feature of webkit.

## License

Copyright (C) 2010 wkhtmltopdf/wkhtmltoimage Authors.

License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>. This is free software: you are free to change and redistribute it. There is NO WARRANTY, to the extent permitted by law.

## Authors

Written by Jan Habermann, Christian Sciberras and Jakob Truelsen. Patches by Mehdi Abbad, Lyes Amazouz, Pascal Bach, Emmanuel Bouthenot, Benoit Garret and Mário Silva.

## Synopsis

```
wkhtmltopdf [GLOBAL OPTION]... [OBJECT]... <output file>
```

## Document objects

wkhtmltopdf is able to put several objects into the output file, an object is either a single webpage, a cover webpage or a table of content. The objects are put into the output document in the order they are specified on the command line, options can be specified on a per object basis or in the global options area. Options from the [Global Options](#) section can only be placed in the global options area

A page objects puts the content of a singe webpage into the output document.

```
(page)? <input url/file name> [PAGE OPTION]...
```

Options for the page object can be placed in the global options and the page options areas. The applicable options can be found in the [Page Options](#) and [Headers And Footer Options](#) sections.

A cover objects puts the content of a single webpage into the output document, the page does not appear in the table of content, and does not have headers and footers.

```
cover <input url/file name> [PAGE OPTION]...
```

All options that can be specified for a page object can also be specified for a cover.

A table of content object inserts a table of content into the output document.

```
toc [TOC OPTION]...
```

All options that can be specified for a page object can also be specified for a toc, further more the options from the [TOC Options](#) section can also be applied. The table of content is generated via XSLT which means that it can be styled to look however you want it to look. To get an aide of how to do this you can dump the default xslt document by supplying the `--dump-default-toc-xsl`, and the outline it works on by supplying `--dump-outline`, see the [Outline Options](#) section.

## Global Options

<code>--collate</code>		Collate when printing multiple copies (default)
<code>--no-collate</code>		Do not collate when printing multiple copies
<code>--cookie-jar</code>	<code>&lt;path&gt;</code>	Read and write cookies from and to the supplied cookie jar file
<code>--copies</code>	<code>&lt;number&gt;</code>	Number of copies to print into the pdf file (default 1)
<code>-d, --dpi</code>	<code>&lt;dpi&gt;</code>	Change the dpi explicitly (this has no effect on X11 based systems)

<b>-H,</b>	<b>--extended-help</b>		Display more extensive help, detailing less common command switches
<b>-g,</b>	<b>--grayscale</b>		PDF will be generated in grayscale
<b>-h,</b>	<b>--help</b>		Display help
	<b>--htmldoc</b>		Output program html help
	<b>--image-dpi*</b>	<i>&lt;integer&gt;</i>	When embedding images scale them down to this dpi (default 600)
	<b>--image-quality*</b>	<i>&lt;integer&gt;</i>	When jpeg compressing images use this quality (default 94)
<b>-I,</b>	<b>--lowquality</b>		Generates lower quality pdf/ps. Useful to shrink the result document space
	<b>--manpage</b>		Output program man page
<b>-B,</b>	<b>--margin-bottom</b>	<i>&lt;unitreal&gt;</i>	Set the page bottom margin (default 10mm)
<b>-L,</b>	<b>--margin-left</b>	<i>&lt;unitreal&gt;</i>	Set the page left margin (default 10mm)
<b>-R,</b>	<b>--margin-right</b>	<i>&lt;unitreal&gt;</i>	Set the page right margin (default 10mm)
<b>-T,</b>	<b>--margin-top</b>	<i>&lt;unitreal&gt;</i>	Set the page top margin (default 10mm)
<b>-O,</b>	<b>--orientation</b>	<i>&lt;orientation&gt;</i>	Set orientation to Landscape or Portrait (default Portrait)
	<b>--output-format</b>	<i>&lt;format&gt;</i>	Specify an output format to use pdf or ps, instead of looking at the extension of the output filename
	<b>--page-height</b>	<i>&lt;unitreal&gt;</i>	Page height
<b>-s,</b>	<b>--page-size</b>	<i>&lt;Size&gt;</i>	Set paper size to: A4, Letter, etc. (default A4)
	<b>--page-width</b>	<i>&lt;unitreal&gt;</i>	Page width
	<b>--no-pdf-compression*</b>		Do not use lossless compression on pdf objects
<b>-q,</b>	<b>--quiet</b>		Be less verbose
	<b>--read-args-from-stdin</b>		Read command line arguments from stdin
	<b>--readme</b>		Output program readme
	<b>--title</b>	<i>&lt;text&gt;</i>	The title of the generated pdf file (The title of the first document is used if not specified)
	<b>--use-xserver*</b>		Use the X server (some plugins and other stuff might not work without X11)
<b>-V,</b>	<b>--version</b>		Output version information and exit

Items marked \* are only available using patched QT.

## Outline Options

<b>--dump-default-toc-xsl*</b>		Dump the default TOC xsl style sheet to stdout
<b>--dump-outline*</b>	<i>&lt;file&gt;</i>	Dump the outline to a file
<b>--outline*</b>		Put an outline into the pdf (default)
<b>--no-outline*</b>		Do not put an outline into the pdf
<b>--outline-depth*</b>	<i>&lt;level&gt;</i>	Set the depth of the outline (default 4)

Items marked \* are only available using patched QT.

## Page Options

<b>--allow</b>	<i>&lt;path&gt;</i>	Allow the file or files from the specified folder to be loaded (repeatable)
<b>--background</b>		Do print background (default)
<b>--no-background</b>		Do not print background
<b>--checkbox-checked-svg</b>	<i>&lt;path&gt;</i>	Use this SVG file when rendering checked checkboxes
<b>--checkbox-svg</b>	<i>&lt;path&gt;</i>	Use this SVG file when rendering unchecked checkboxes
<b>--cookie</b>	<i>&lt;name&gt; &lt;value&gt;</i>	Set an additional cookie (repeatable)
<b>--custom-header</b>	<i>&lt;name&gt; &lt;value&gt;</i>	Set an additional HTTP header (repeatable)
<b>--custom-header-propagation</b>		Add HTTP headers specified by --custom-header for each resource request.
<b>--no-custom-header-propagation</b>		Do not add HTTP headers specified by --custom-header for each resource request.
<b>--debug-javascript</b>		Show javascript debugging output
<b>--no-debug-javascript</b>		Do not show javascript debugging output (default)
<b>--default-header*</b>		Add a default header, with the name of the page to the left, and the page number to the right, this is short for: --header-left="[webpage]" --header-right="[page]/[toPage]" --top 2cm --header-line
<b>--encoding</b>	<i>&lt;encoding&gt;</i>	Set the default text encoding, for input
<b>--disable-external-links*</b>		Do not make links to remote web pages
<b>--enable-external-links*</b>		Make links to remote web pages (default)
<b>--disable-forms*</b>		Do not turn HTML form fields into pdf form fields (default)
<b>--enable-forms*</b>		Turn HTML form fields into pdf form fields
<b>--images</b>		Do load or print images (default)
<b>--no-images</b>		Do not load or print images
<b>--disable-internal-links*</b>		Do not make local links
<b>--enable-internal-links*</b>		Make local links (default)
<b>-n,</b>	<b>--disable-javascript</b>	Do not allow web pages to run javascript
	<b>--enable-javascript</b>	Do allow web pages to run javascript (default)
	<b>--javascript-delay</b>	<i>&lt;msec&gt;</i> Wait some milliseconds for javascript finish (default 200)
	<b>--load-error-handling</b>	<i>&lt;handler&gt;</i> Specify how to handle pages that fail to load: abort, ignore or skip (default abort)
	<b>--disable-local-file-access</b>	Do not allowed conversion of a local file to read in other local files, unless explicitly allowed with --allow

<b>--enable-local-file-access</b>		Allowed conversion of a local file to read in other local files. (default)
<b>--minimum-font-size</b>	<code>&lt;int&gt;</code>	Minimum font size
<b>--exclude-from-outline*</b>		Do not include the page in the table of contents and outlines
<b>--include-in-outline*</b>		Include the page in the table of contents and outlines (default)
<b>--page-offset</b>	<code>&lt;offset&gt;</code>	Set the starting page number (default 0)
<b>--password</b>	<code>&lt;password&gt;</code>	HTTP Authentication password
<b>--disable-plugins</b>		Disable installed plugins (default)
<b>--enable-plugins</b>		Enable installed plugins (plugins will likely not work)
<b>--post</b>	<code>&lt;name&gt; &lt;value&gt;</code>	Add an additional post field (repeatable)
<b>--post-file</b>	<code>&lt;name&gt; &lt;path&gt;</code>	Post an additional file (repeatable)
<b>--print-media-type*</b>		Use print media-type instead of screen
<b>--no-print-media-type*</b>		Do not use print media-type instead of screen (default)
<b>-p, --proxy</b>	<code>&lt;proxy&gt;</code>	Use a proxy
<b>--radiobutton-checked-svg</b>	<code>&lt;path&gt;</code>	Use this SVG file when rendering checked radiobuttons
<b>--radiobutton-svg</b>	<code>&lt;path&gt;</code>	Use this SVG file when rendering unchecked radiobuttons
<b>--run-script</b>	<code>&lt;js&gt;</code>	Run this additional javascript after the page is done loading (repeatable)
<b>--disable-smart-shrinking*</b>		Disable the intelligent shrinking strategy used by WebKit that makes the pixel/dpi ratio none constant
<b>--enable-smart-shrinking*</b>		Enable the intelligent shrinking strategy used by WebKit that makes the pixel/dpi ratio none constant (default)
<b>--stop-slow-scripts</b>		Stop slow running javascripts (default)
<b>--no-stop-slow-scripts</b>		Do not Stop slow running javascripts (default)
<b>--disable-toc-back-links*</b>		Do not link from section header to toc (default)
<b>--enable-toc-back-links*</b>		Link from section header to toc
<b>--user-style-sheet</b>	<code>&lt;url&gt;</code>	Specify a user style sheet, to load with every page
<b>--username</b>	<code>&lt;username&gt;</code>	HTTP Authentication username
<b>--window-status</b>	<code>&lt;windowStatus&gt;</code>	Wait until window.status is equal to this string before rendering page
<b>--zoom</b>	<code>&lt;float&gt;</code>	Use this zoom factor (default 1)

Items marked \* are only available using patched QT.

## Headers And Footer Options

<b>--footer-center*</b>	<code>&lt;text&gt;</code>	Centered footer text
<b>--footer-font-name*</b>	<code>&lt;name&gt;</code>	Set footer font name (default Arial)
<b>--footer-font-size*</b>	<code>&lt;size&gt;</code>	Set footer font size (default 12)
<b>--footer-html*</b>	<code>&lt;url&gt;</code>	Adds a html footer
<b>--footer-left*</b>	<code>&lt;text&gt;</code>	Left aligned footer text
<b>--footer-line*</b>		Display line above the footer
<b>--no-footer-line*</b>		Do not display line above the footer (default)
<b>--footer-right*</b>	<code>&lt;text&gt;</code>	Right aligned footer text
<b>--footer-spacing*</b>	<code>&lt;real&gt;</code>	Spacing between footer and content in mm (default 0)
<b>--header-center*</b>	<code>&lt;text&gt;</code>	Centered header text
<b>--header-font-name*</b>	<code>&lt;name&gt;</code>	Set header font name (default Arial)
<b>--header-font-size*</b>	<code>&lt;size&gt;</code>	Set header font size (default 12)
<b>--header-html*</b>	<code>&lt;url&gt;</code>	Adds a html header
<b>--header-left*</b>	<code>&lt;text&gt;</code>	Left aligned header text
<b>--header-line*</b>		Display line below the header
<b>--no-header-line*</b>		Do not display line below the header (default)
<b>--header-right*</b>	<code>&lt;text&gt;</code>	Right aligned header text
<b>--header-spacing*</b>	<code>&lt;real&gt;</code>	Spacing between header and content in mm (default 0)
<b>--replace*</b>	<code>&lt;name&gt; &lt;value&gt;</code>	Replace [name] with value in header and footer (repeatable)

Items marked \* are only available using patched QT.

## TOC Options

<b>--disable-dotted-lines*</b>		Do not use dotted lines in the toc
<b>--toc-header-text*</b>	<code>&lt;text&gt;</code>	The header text of the toc (default Table of Content)
<b>--toc-level-indentation*</b>	<code>&lt;width&gt;</code>	For each level of headings in the toc indent by this length (default 1em)
<b>--disable-toc-links*</b>		Do not link from toc to sections
<b>--toc-text-size-shrink*</b>	<code>&lt;real&gt;</code>	For each level of headings in the toc the font is scaled by this factor (default 0.8)
<b>--xsl-style-sheet*</b>	<code>&lt;file&gt;</code>	Use the supplied xsl style sheet for printing the table of content

Items marked \* are only available using patched QT.

## Specifying A Proxy

By default proxy information will be read from the environment variables: proxy, all\_proxy and http\_proxy, proxy options can also be specified with the -p switch

```
<type> := "http://" | "socks5://"
<serif> := <username> (":" <password>)? "@"
<proxy> := "None" | <type>? <serif>? <host> (":" <port>)?
```

Here are some examples (In case you are unfamiliar with the BNF):

```
http://user:password@myproxyserver:8080
socks5://myproxyserver
None
```

## Footers And Headers

Headers and footers can be added to the document by the `--header-*` and `--footer*` arguments respectfully. In header and footer text string supplied to e.g. `--header-left`, the following variables will be substituted.

```
* [page]      Replaced by the number of the pages currently being printed
* [frompage]  Replaced by the number of the first page to be printed
* [topage]    Replaced by the number of the last page to be printed
* [webpage]   Replaced by the URL of the page being printed
* [section]   Replaced by the name of the current section
* [subsection] Replaced by the name of the current subsection
* [date]      Replaced by the current date in system local format
* [time]      Replaced by the current time in system local format
* [title]     Replaced by the title of the of the current page object
* [doctitle]  Replaced by the title of the output document
```

As an example specifying `--header-right "Page [page] of [toPage]"`, will result in the text "Page x of y" where x is the number of the current page and y is the number of the last page, to appear in the upper left corner in the document.

Headers and footers can also be supplied with HTML documents. As an example one could specify `--header-html header.html`, and use the following content in `header.html`:

```
<html><head><script>
function subst() {
  var vars={};
  var x=document.location.search.substring(1).split('&');
  for (var i in x) {var z=x[i].split('=');vars[z[0]] = unescape(z[1]);}
  var x=['frompage','topage','page','webpage','section','subsection','subsubsection'];
  for (var i in x) {
    var y = document.getElementsByClassName(x[i]);
    for (var j=0; j<y.length; ++j) y[j].textContent = vars[x[i]];
  }
}
</script></head><body style="border:0; margin: 0;" onload="subst()">
<table style="border-bottom: 1px solid black; width: 100%">
  <tr>
    <td class="section"></td>
    <td style="text-align:right">
      Page <span class="page"></span> of <span class="topage"></span>
    </td>
  </tr>
</table>
</body></html>
```

As can be seen from the example, the arguments are sent to the header/footer html documents in get fashion.

## Outlines

Wkhtmltopdf with patched qt has support for PDF outlines also known as book marks, this can be enabled by specifying the `--outline` switch. The outlines are generated based on the `<h?>` tags, for a in-depth description of how this is done see the [Table Of Contest](#) section.

The outline tree can sometimes be very deep, if the `<h?>` tags where spread to generous in the HTML document. The `--outline-depth` switch can be used to bound this.

## Table Of Content

A table of content can be added to the document by adding a toc object to the command line. For example:

```
wkhtmltopdf toc http://doc.trolltech.com/4.6/qstring.html qstring.pdf
```

The table of content is generated based on the H tags in the input documents. First a XML document is generated, then it is converted to HTML using XSLT.

The generated XML document can be viewed by dumping it to a file using the `--dump-outline` switch. For example:

```
wkhtmltopdf --dump-outline toc.xml http://doc.trolltech.com/4.6/qstring.html qstring.pdf
```

The XSLT document can be specified using the `--xsl-style-sheet` switch. For example:

```
wkhtmltopdf toc --xsl-style-sheet my.xsl http://doc.trolltech.com/4.6/qstring.html qstring.pdf
```

The `--dump-default-toc-xsl` switch can be used to dump the default XSLT style sheet to stdout. This is a good start for writing your own style sheet

```
wkhtmltopdf --dump-default-toc-xsl
```

The XML document is in the namespace "http://code.google.com/p/wkhtmltopdf/outline" it has a root node called "outline" which contains a number of "item" nodes. An item can contain any number of item. These are the outline subsections to the section the item represents. A item node has the following attributes:

- "title" the name of the section.
- "page" the page number the section occurs on.
- "link" a URL that links to the section.

- "backLink" the name of the anchor the the section will link back to.

The remaining TOC options only affect the default style sheet so they will not work when specifying a custom style sheet.

## Page Breaking

The current page breaking algorithm of WebKit leaves much to be desired. Basically webkit will render everything into one long page, and then cut it up into pages. This means that if you have two columns of text where one is vertically shifted by half a line. Then webkit will cut a line into to pieces display the top half on one page. And the bottom half on another page. It will also break image in two and so on. If you are using the patched version of QT you can use the CSS page-break-inside property to remedy this somewhat. There is no easy solution to this problem, until this is solved try organising your HTML documents such that it contains many lines on which pages can be cut cleanly.

See also: <http://code.google.com/p/wkhtmltopdf/issues/detail?id=9>, <http://code.google.com/p/wkhtmltopdf/issues/detail?id=33> and <http://code.google.com/p/wkhtmltopdf/issues/detail?id=57>.

## Page sizes

The default page size of the rendered document is A4, but using this `--page-size` option this can be changed to almost anything else, such as: A3, Letter and Legal. For a full list of supported pages sizes please see <http://doc.trolltech.com/4.6/qprinter.html#PageSize-enum>.

For a more fine grained control over the page size the `--page-height` and `--page-width` options may be used

## Reading arguments from stdin

If you need to convert a lot of pages in a batch, and you feel that wkhtmltopdf is a bit to slow to start up, then you should try `--read-args-from-stdin`,

When `--read-args-from-stdin` each line of input sent to wkhtmltopdf on stdin will act as a separate invocation of wkhtmltopdf, with the arguments specified on the given line combined with the arguments given to wkhtmltopdf

For example one could do the following:

```
echo "http://doc.trolltech.com/4.5/qapplication.html qapplication.pdf" >> cmds
echo "cover google.com http://en.wikipedia.org/wiki/Qt_(toolkit) qt.pdf" >> cmds
wkhtmltopdf --read-args-from-stdin --book < cmds
```

## Static version

On the wkhtmltopdf website you can download a static version of wkhtmltopdf <http://code.google.com/p/wkhtmltopdf/downloads/list>. This static binary will work on most systems and comes with a build in patched QT.

Unfortunately the static binary is not particularly static, on Linux it depends on both glibc and openssl, furthermore you will need to have an xserver installed but not necessary running. You will need to have different fonts install including xfonts-scalable (Type1), and msttcfonts. See <http://code.google.com/p/wkhtmltopdf/wiki/static> for trouble shooting.

## Compilation

It can happen that the static binary does not work for your system for one reason or the other, in that case you might need to compile wkhtmltopdf yourself.

### GNU/Linux:

Before compilation you will need to install dependencies: X11, gcc, git and openssl. On Debian/Ubuntu this can be done as follows:

```
sudo apt-get build-dep libqt4-gui libqt4-network libqt4-webkit
sudo apt-get install openssl build-essential xorg git-core git-doc libssl-dev
```

On other systems you must use your own package manager, the packages might be named differently.

First you must check out the modified version of QT

```
git clone git://gitorious.org/wkhtml2pdf/qt/wkhtmltopdf-qt.git wkhtmltopdf-qt
```

Next you must configure, compile and install QT, note this will take quite some time, depending on what arguments you use to configure qt

```
cd wkhtmltopdf-qt
./configure -nomake tools,examples,demos,docs,translations -opensource -prefix ../wkqt
make -j3
make install
cd ..
```

All that is needed now is, to compile wkhtmltopdf.

```
git clone git://github.com/antialize/wkhtmltopdf.git wkhtmltopdf
cd wkhtmltopdf
../wkqt/bin/qmake
make -j3
```

You now have a binary called wkhtmltopdf in the currently folder that you can use, you can optionally install it by running

```
make install
```

### Other operative systems and advanced features

If you want more details or want to compile under other operative systems other than GNU/Linux, please see <http://code.google.com/p/wkhtmltopdf/wiki/compilation>.

## Installation

There are several ways to install wkhtmltopdf. You can download a already compiled binary, or you can compile wkhtmltopdf yourself. On windows the easiest way to install wkhtmltopdf is to download the latest installer. On linux you can download the latest static binary, however you still need to install some other pieces of software, to learn more about this read the static version section of the manual.

## Examples

This section presents a number of examples of how to invoke wkhtmltopdf.

To convert a remote HTML file to PDF:

```
wkhtmltopdf http://www.google.com google.pdf
```

To convert a local HTML file to PDF:

```
wkhtmltopdf my.html my.pdf
```

You can also convert to PS files if you like:

```
wkhtmltopdf my.html my.ps
```

Produce the eler2.pdf sample file:

```
wkhtmltopdf -H http://geekz.co.uk/lovesraymond/archive/eler-highlights-2008 eler2.pdf
```

Printing a book with a table of content:

```
wkhtmltopdf -H cover cover.html toc chapter1.html chapter2.html chapter3.html book.pdf
```