**pdf2htmlEX**

[Build Status](https://travis-ci.org/coolwanglu/pdf2htmlEX)

A beautiful demo is worth a thousand words:

* **Typography** (fonts): [HTML](http://coolwanglu.github.com/pdf2htmlEX/demo/geneve.html) / [PDF](https://github.com/raphink/geneve_1564/raw/master/geneve_1564.pdf)
* **Cheat Sheet** (math formulas): [HTML](http://coolwanglu.github.com/pdf2htmlEX/demo/cheat.html) / [PDF](http://www.tug.org/texshowcase/cheat.pdf)
* **Scientific Paper** (text and figures): [HTML](http://coolwanglu.github.com/pdf2htmlEX/demo/demo.html) / [PDF](http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.148.349&rep=rep1&type=pdf)
* **Full Circle Magazine** (read while downloading): [HTML](http://coolwanglu.github.com/pdf2htmlEX/demo/issue65_en.html) / [PDF](http://dl.fullcirclemagazine.org/issue65_en.pdf)
* **Git Manual** (CJK support): [HTML](http://coolwanglu.github.com/pdf2htmlEX/demo/chn.html) / [PDF](http://files.cnblogs.com/phphuaibei/git%E6%90%AD%E5%BB%BA.pdf)
* [Try your own files](https://github.com/coolwanglu/pdf2htmlEX/wiki/UploadDemo)

[Browser requirements](https://github.com/coolwanglu/pdf2htmlEX/wiki/Browser-Requirements)

**Introduction**

pdf2htmlEX renders PDF files in HTML, utilizing modern Web technologies. It aims to provide an accurate rendering, while keeping optimized for Web display.

pdf2htmlEX is best for text-based PDF files, for example scientific papers with complicated formulas and figures. Text, fonts and formats are natively preserved in HTML such that you can still search and copy. The generated HTML file is static, with optional features powered by JavaScript.

[Learn more about who and why should use pdf2htmlEX](https://github.com/coolwanglu/pdf2htmlEX/wiki/Introduction)

**Features**

* Precise and native text in HTML
* Flexible Output
* Moderate Size
* More PDF stuffs that you love: links, outlines & printing

[Learn more](https://github.com/coolwanglu/pdf2htmlEX/wiki/Feature-List)  
[Compare with others](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison)

**Wiki Portals**

* [Wiki Home](https://github.com/coolwanglu/pdf2htmlEX/wiki)
* [Download](https://github.com/coolwanglu/pdf2htmlEX/wiki/Download)
* [Building](https://github.com/coolwanglu/pdf2htmlEX/wiki/Building)
* [Quick Start](https://github.com/coolwanglu/pdf2htmlEX/wiki/QuickStart)
* [How to contribute](https://github.com/coolwanglu/pdf2htmlEX/wiki/Contribute)
* [FAQ](https://github.com/coolwanglu/pdf2htmlEX/wiki/FAQ)

**LICENSE**

GPLv3 with additional terms (see below) for most parts, MIT License for share/\*

Read LICENSE for more detail.

**For Online Services**

You are free and welcome to modify pdf2htmlEX for your online services, but you should credit pdf2htmlEX if your service involves "online conversion" facilitated by pdf2htmlEX. You are also encouraged to send me a name and a URL for the purpose of statistics.

Read LICENSE for more detail.

**Resources**

* [pdf2htmlEX Blog](http://pdf2htmlex.blogspot.com/)
  + News and announcements
  + [RSS Subscribe](http://pdf2htmlex.blogspot.com/feeds/posts/default)
* Mailing list [pdf2htmlex@googlegroups.com](mailto:pdf2htmlex@googlegroups.com)
  + Questions are welcome, but make sure you have read man pdf2htmlEX and [wiki](https://github.com/coolwanglu/pdf2htmlEX/wiki)
  + [Subscribe](https://groups.google.com/forum/#!forum/pdf2htmlex)
* Lu Wang [coolwanglu@gmail.com](mailto:coolwanglu@gmail.com) or [coolwanglu@twitter](https://twitter.com/coolwanglu)
  + For personal enquiries only
  + Accepting messages in **中文**, **English** or **日本語**
* IRC Channel irc://irc.freenode.net/pdf2htmlEX

**Acknowledgements**

pdf2htmlEX is made possible thanks to the following projects:

* [poppler](http://poppler.freedesktop.org/)
* [Fontforge](http://fontforge.org/)
* [jQuery](http://jquery.com/)

pdf2htmlEX is inspired by the following projects:

* pdftops & pdftohtml from poppler
* MuPDF
* PDF.js
* Crocodoc
* Google Doc

**Special Thanks**

* Hongliang Tian
* Wanmin Liu

# Browser Requirements

A few features are required for a browser to show HTML files generated by pdf2htmlEX correctly

* HTML5: A basic support is enough, especially the DOCTYPE.
  + Advanced features like <canvas>, <video> etc are NOT required.
* CSS: Essential functional part.
  + @font-face, supports TrueType at least
  + data URI
  + transformation
  + absolute positioning
* Javascript: only required by the default UI, which is optional
  + see also the requirements of jQuery. The default UI relies on a few functions of jQuery.

A recent version of Firefox or Chrome, or IE(>=9) should work.

### Notes

* [MacType](http://code.google.com/p/mactype/) might help if you are using Chrome on Windows.
* Turn on ClearType if you are using Windows XP: (Control Panel -> Display -> Appearance -> Effects -> "Use the following method to smooth edges of screen fonts" -> ClearType)

Last edited by Lu Wang, 3 months ago

**Comparison**

This page compares common approaches to present PDF files online.

**Basic Info**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Convert to HTML 5** | **Parse by JS** | **Convert to image** | **Convert toHTML 4** | **AdobePDFplugin** | **Other plugins** |
| Example | pdf2htmlEX | PDF.js | pdftoppm (poppler) Google Doc | pdftohtml (poppler) | AdobePDFPlugin | N/A |
| Briefing | PDF elements are converted into corresponding or closest HTMLelements | PDF file is loaded, parsed and rendered by Javascript | PDF pages are converted into images and shown in web pages | Similar as “Convert toHTML 5”, but with much less features | Official plugin | Non-officialPDFplugins, Flash-based plugins or others |
| Open source | Yes | Some (pdf.js) | Poppler is open source. Google Doc may be based on poppler as well, because they showed same errors. | Some (pdftohtml) | No | Maybe |
| Free | Yes | Some | Some | Some | Yes | Some |

note: There are free and/or open source tools for all but Adobe PDF plugin.

**Performance**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Convert to HTML 5 (pdf2htmlEX)** | **Parse by JS** | **Convert to image** | **Convert toHTML 4** | **Adobe PDFplugin** | **Other plugins** |
| Processing (server-side) | Normal, one time | None | Slow, one time | Fast, one time | None | None, usually |
| Loading (client-side) | Fast | Fast | Slow | Fast | Fast | Fast |
| Rendering (client-side) | Fast | Slow | Fast | Fast | Fast | Fast, usually |
| Network cost | Small [1](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn1) | Small | Large [2](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn2) | Small | Small | Small |

[1](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn1): HTTP compression is required  
[2](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn2): Could be Huge if higher resolution is needed

**Browser Requirements**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Convert to HTML 5 (pdf2htmlEX)** | **Parse by JS** | **Convert to image** | **Convert toHTML 4** | **Adobe PDFplugin** | **Other plugins** |
| HTML 5 | Yes | Yes, usually | No | No | No | No |
| CSS | Yes | Yes | No | Yes | No | No |
| Javascript | No | Yes | No | No | No | No |
| Third-party plugin | No | No | No | No | Yes | Yes |

**Features**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Convert to HTML 5 (pdf2htmlEX)** | **Parse by JS** | **Convert to image** | **Convert toHTML 4** | **Adobe PDFplugin** | **Other plugins** |
| Full PDF Feature ? | No, but usually enough | Maybe | Yes | No | Yes | Maybe |
| Text Extraction (select/copy/search) | Yes | Yes, with text layer | No, usually [1](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn1) | Yes | Yes | Maybe |
| Link | Yes | Yes | No, usually [2](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn2) | Yes | Yes | Maybe |
| Accurate rendering | Yes, usually [3](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn3) | Yes | Yes | No | Yes | Yes, usually |
| Read while loading | Yes | Yes | Yes | Yes | No | Maybe |

[1](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn1): Text extraction can be supported with a text layer  
[2](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn2): Link may be handled with Javascript  
[3](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn3): There are PDF elements which cannot be converted into HTML losslessly

**Development**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **Convert to HTML 5 (pdf2htmlEX)** | **Parse by JS** | **Convert to image** | **Convert toHTML 4** | **Adobe PDFplugin** | **Other plugins** |
| Customizable UI/Theme | Yes | Yes | Yes | Yes | No | No, usually[1](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn1) |
| Extensible | Yes | Yes | Yes | Yes | No | Maybe [2](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn2) |

[1](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn1): For some plugins there are commercial licensed versions with customizable UI  
[2](https://github.com/coolwanglu/pdf2htmlEX/wiki/Comparison#wiki-fn2): Some plugins have API available

Last edited by shenzhuxi, 3 months ago

# Feature List

### Features

* Accurate Text preservation
  + Native text in HTML - you can select & copy & search
  + Font, color, position and transformation
  + Reencoding for Web
  + Fallback mode (image + hidden text), provides even better accuracy and compatibility, at the cost of larger file sizes.
* Flexible Output
  + Different modes
  + Normal single HTML, with separate fonts/images.
  + All-in-one HTML, with all fonts/images embeded.
  + One HTML file per page, best for dynamic web pages.
  + Appearance can be easily customized with CSS.
  + Content can be accessed with JavaScript.
* Moderate Size
  + Usually similar as original PDF file uncompressed.
* Links
  + In-document reference
  + With Javascript: precise jump
  + Without Javascript: jump to page
  + Out-document reference
* Outline
* Printing (experimental)
* Rectangular Clipping (experimental)

### Working in progress

* CSS tricks
  + Simple path drawing, lines, rectangles...
  + Linear gradients
* Partially supported (All rendered into background images)
  + Type 3 fonts
  + Path paintings/Images

Last edited by Lu Wang, 3 months ago

# Building

#### Dependencies

* CMake, pkg-config
* GNU Getopt
* compilers support C++11, for example
  + GCC >= 4.4.6
  + I heard about successful build with Clang
* **poppler** >= 0.20.0 with xpdf headers (compile with **--enable-xpdf-headers**)
  + Install **libpng** (and headers) BEFORE you compile poppler if you want background images generated
  + Install **poppler-data** if your want CJK support
* **fontforge** (with header files)
  + git version is recommended
* [Optional] **ttfautohint**
  + run pdf2htmlEX with **--external-hint-tool=ttfautohint** to enable it
* [For Windows]
  + Cygwin
  + or MinGW, with some modifications to pdf2htmlEX. See [pdf2htmlEX on TeX Wiki](http://oku.edu.mie-u.ac.jp/~okumura/texwiki/?pdf2htmlEX) (in Japanese), special thanks to Haruhiko Okumura

#### Compiling

git clone git://github.com/coolwanglu/pdf2htmlEX.git

cd pdf2htmlEX

cmake . && make && sudo make install

#### Troubleshooting

If you see error messages about:

* goo/GooString.h, read the dependencies again, poppler should be compiled with --enable-xpdf-headers
* spiroentrypoints.h, install header files of libspiro
* undefined reference of Py\_xxx, install header files of python-2.x
* libintl.h, install gettext and set your system include path accordingly.
* .../libfontforge.so: undefined reference to ..., it should be caused by [a bug of FontForge](https://github.com/fontforge/fontforge/issues/465), please leave comments there such that FontForge developers may fix it soon.
  + You may try [the tmp branch of my fork](https://github.com/coolwanglu/fontforge/tree/tmp)
  + Try to configure with --without-libzmq --without-x --without-iconv --disable-python-scripting --disable-python-extension

Last edited by Lu Wang, 3 months ago

# QuickStart

### This page lists some common recipes of pdf2htmlEX

### First thing first

* It is highly recommended that you install ttfautohint and always add --external-hint-tool=ttfautohint to each of the following recipes. This tool enhances font rendering for all browsers on Windows.
* Double check you have poppler-data installed, for CJK characters.
* Double check you have run sudo make install, or pdf2htmlEX may not be executed correctly

### The simplest case

Suppose you have a PDF file pdf/test.pdf, simply running

pdf2htmlEX --zoom 1.3 pdf/test.pdf

would produce a single HTML file test.html in the **current directory**.

### Advanced

pdf2htmlEX -f 3 -l 5 --fit-width 1024 pdf/test.pdf

would convert only the 3rd, 4th and 5th pages, and fit the page width to 1024 pixels

### For publishers

pdf2htmlEX --embed cfijo --dest-dir out pdf/test.pdf

would produce a test.html and accompanying files in the out directory, in this way all the resources (fonts, images, css and javascript) are stored in separated files such that the viewer can take more advantage of browser caches.

### For advanced publishers

pdf2htmlEX --embed cfijo --split-pages 1 --dest-dir out --page-filename test-%d.page pdf/test.pdf

would do something similar above, but each individual page is stored in a separated file. The files are named as test-0.page, test-1.pageand so on, as specified in the command line. There is still a test.html which loads the pages dynamically through ajax. In this way the publishers are given full control, who can organize the pages as they like, for example, to implement lazy page loading.

### The Ultimate Hand

pdf2htmlEX --fallback 1 pdf/test.pdf

would also produce a single test.html, which, however, consists of images and hidden text. This mode provides maximum accuracy and compatibility, at the cost of larger file size. Use this mode only when pdf2htmlEX cannot correctly process your files otherwise.

### More

Just remember man pdf2htmlEX and pdf2htmlEX --help are always your best friends.

Last edited by Jason Moon, 2 months ago

# FAQ

# dummy

## Non-technical

### Languages and libraries used in pdf2htmlEX

* C++ (most part)
* C (wrapper of Fontforge)
* HTML (output)
* CSS (complicated enough to be considered as a language)
* Javascript (UI actions/effects)
* Python (scripts for testing / packaging)
* Poppler (PDF parsing)
* Fontforge (font manipulation)
* jQuery (for the default UI)

### pdf2htmlEX doesn't work for my pdf file

* Bug reports are always welcome, please file an issue with the link to the broken pdf file.
* However there are several exceptions when the bug cannot be fixed in time (or at all)
  + The file does not follow the PDF standard (it might still be displayed correctly in PDF viewers)
  + Something wrong with libraries used by pdf2htmlEX (poppler / fontforge)
  + There are a few technical limitations of pdf2htmlEX. See [this page](https://github.com/coolwanglu/pdf2htmlEX/wiki/Limitations)

### I want more features!

* Create a patch, or hire someone to do so.
* Best hackers do not work for free.
* But great ideas are more valuable than money.

### How to install pdf2htmlEX on Windows

* There is no binary package so far, so do not expect an easy click-and-install. There are 2 options:
  + Install Linux in a virtual machine, then install pdf2htmlEX
  + Build the source through Cygwin (recommended) or MinGW
* If you can help maintain binary packages for Windows, or at least find an easy way to do this, please contact me.

## Pitfalls

### 'Cannot open the manifest file'

* Run 'sudo make install' or 'make install', depending on your environment.

### There is no image generated

* Make sure you did not specify --process-nontext 0
* Make sure libpng (and headers) is installed BEFORE poppler was compiled.

### The generated HTML file looks ugly

Check if your browser meets the [requirements](https://github.com/coolwanglu/pdf2htmlEX/wiki/Browser-Requirements).

### The generated HTML file freezes my Firefox

* Don't zoom in too much
* Use a smaller value for --font-size-multiplier

### Something wrong with text in the generated HTML file, I cannot even read them

* Install ttfautohint and run pdf2htmlEX with --external-hint-tool=ttfautohint
* Try --auto-hint 1 carefully, which is experimental now.

### I got incorrect text after copy & paste

* try run with --tounicode 1
* Make sure you CAN copy & paste with a PDF viewer

### Generated text are too small to read

* try run with --zoom 2

### Images are blurred

* try run with --hdpi 288 --vdpi 288

Last edited by Lu Wang, 3 months ago

# Command line

This is the man page of version 0.9. Run man pdf2htmlEX for the latest manual.

Please file an issue if you find this page outdated.

NAME

pdf2htmlEX - converts PDF to HTML without losing text and format.

USAGE

pdf2htmlEX [options] <input-filename> [<output-filename>]

DESCRIPTION

pdf2htmlEX is a utility that converts PDF files to HTML files.

pdf2htmlEX tries its best to render the PDF precisely, maintain proper styling, while retaining text and optimizing for Web.

Fonts are extracted form PDF and then embedded into HTML (Type 3 fonts are not supported). Text in the converted HTML file is usually selectable and

copyable.

Other objects are rendered as images and also embedded.

OPTIONS

Pages

-f, --first-page <num> (Default: 1)

Specify the first page to process

-l, --last-page <num> (Default: last page)

Specify the last page to process

Dimensions

--zoom <ratio>, --fit-width <width>, --fit-height <height>

--zoom specifies the zoom factor directly; --fit-width/height specifies the maximum width/height of a page, the values are in pixels.

If multiple values are specified, the minimum one will be used.

If none is specified, pages will be rendered as 72DPI.

--use-cropbox <0|1> (Default: 1)

Use CropBox instead of MediaBox for output.

--hdpi <dpi>, --vdpi <dpi> (Default: 144)

Specify the horizontal and vertical DPI for images

Output

--embed <string>

--embed-css <0|1> (Default: 1)

--embed-font <0|1> (Default: 1)

--embed-image <0|1> (Default: 1)

--embed-javascript <0|1> (Default: 1)

--embed-outline <0|1> (Default: 1)

Specify which elements should be embedded into the output HTML file.

If switched off, separated files will be generated along with the HTML file for the corresponding elements.

--embed accepts a string as argument. Each letter of the string must be one of `cCfFiIjJoO`, which corresponds to one of the --embed-\*\*\*

switches. Lower case letters for 0 and upper case letters for 1. For example, `--embed cFIJo` means to embed everything but CSS files and out‐

lines.

--split-pages <0|1> (Default: 0)

If turned on, the content of each page is stored in a separated file.

--page-filename may be used to specify the format for the filenames for individual pages. A %d placeholder may be included to indicate where

the page number should be placed. The placeholder supports a limited subset of normal numerical placeholders, including specified width and

zero padding.

If --page-filename does not contain a placeholder for the page number, the page number will be inserted directly before the file extension. If

the filename does not have an extension, the page number will be placed at the end of the file name.

If --page-filename is not specified, <input-filename> will be used for the output filename, replacing the extension with .page and adding the

page number directly before the extension.

This switch is useful if you want pages to be loaded separately & dynamically -- a supporting server might be necessary.

Examples

pdf2htmlEX --split-pages 1 foo.pdf

Yields page files foo1.page, foo2.page, etc.

pdf2htmlEX --split-pages 1 foo.pdf bar.baz

Yields page files bar1.baz, bar2.baz, etc.

pdf2htmlEX --split-pages 1 foo.pdf page%dbar.baz

Yields page files page1bar.baz, page2bar.baz, etc.

pdf2htmlEX --split-pages 1 foo.pdf bar%03d.baz

Yields page files bar001.baz, bar002.baz, etc.

--dest-dir <dir> (Default: .)

Specify destination folder.

--css-filename <filename> (Default: <none>)

Specify the filename of the generated css file, if not embedded.

If it's empty, the file name will be determined automatically.

--page-filename <filename> (Default: <none>)

Specify the filename template for pages. This is only useful when --split-pages is 1

If it's empty, a default one will be used, see description of --split-pages

--outline-filename <filename> (Default: <none>)

Specify the filename of the generated outline file, if not embedded.

If it's empty, the file name will be determined automatically.

--process-nontext <0|1> (Default: 1)

Whether to process non-text objects (as images)

--process-outline <0|1> (Default: 1)

Whether to show outline in the generated HTML

--printing <0|1> (Default: 1)

Enable printing support. Disabling this option may reduce the size of CSS.

--fallback <0|1> (Deafult: 0)

Output in fallback mode, for better accuracy and browser compatibility, but the size becomes larger.

Fonts

--embed-external-font <0|1> (Default: 1)

Specify whether the local matched fonts, for fonts not embedded in PDF, should be embedded into HTML.

If this switch is off, only font names are exported such that web browsers may try to find proper fonts themselves, and that might cause issues

about incorrect font metrics.

--font-suffix <suffix> (Default: .ttf)

Specify the suffix of fonts extracted from the PDF file.

--decompose-ligature <0|1> (Default: 0)

Decompose ligatures. For example 'fi' -> 'f''i'.

--auto-hint <0|1> (Default: 0)

If set to 1, hints will be generated for the fonts using fontforge.

This may be preceded by --external-hint-tool.

--external-hint-tool <tool> (Default: <none>)

If specified, the tool will be called in order to enhanced hinting for fonts, this will precede --auto-hint.

The tool will be called as '<tool> <in.suffix> <out.suffix>', where suffix will be the same as specified for --font-suffix.

--stretch-narrow-glyph <0|1> (Default: 0)

If set to 1, glyphs narrower than described in PDF will be stretched; otherwise space will be padded to the right of the glyphs

--squeeze-wide-glyph <0|1> (Default: 1)

If set to 1, glyphs wider than described in PDF will be squeezed; otherwise it will be truncated.

Text

--heps <len>, --veps <len> (Default: 1)

Specify the maximum tolerable horizontal/vertical offset (in pixels).

pdf2htmlEX would try to optimize the generated HTML file moving Text within this distance.

--space-threshold <ratio> (Default: 0.125)

pdf2htmlEX would insert a whitespace character ' ' if the distance between two consecutive letters in the same line is wider than ratio \*

font\_size.

--font-size-multiplier <ratio> (Default: 4.0)

Many web browsers limit the minimum font size, and many would round the given font size, which results in incorrect rendering.

Specify a ratio greater than 1 would resolve this issue, however it might freeze some browsers.

For some versions of Firefox, however, there will be a problem when the font size is too large, in which case a smaller value should be speci‐

fied here.

--space-as-offset <0|1> (Default: 0)

If set to 1, space characters will be treated as offsets, which allows a better optimization.

For PDF files with bad encodings, turning on this option may cause losing characters.

--tounicode <-1|0|1> (Default: 0)

A ToUnicode map may be provided for each font in PDF which indicates the 'meaning' of the characters. However often there is better "ToUnicode"

info in Type 0/1 fonts, and sometimes the ToUnicode map provided is wrong. If this value is set to 1, the ToUnicode Map is always applied, if

provided in PDF, and characters may not render correctly in HTML if there are collisions.

If set to -1, a customized map is used such that rendering will be correct in HTML (visually the same), but you may not get correct characters

by select & copy & paste.

If set to 0, pdf2htmlEX would try its best to balance the two methods above.

--optimize-text <0|1> (Deafult: 0)

If set to 1, pdf2htmlEX will try to reduce the number of HTML elements used for text. Turn it off if anything goes wrong.

PDF Protection

-o, --owner-password <password>

Specify owner password

-u, --user-password <password>

Specify user password

--no-drm <0|1> (Default: 0)

Override document DRM settings

Misc.

--clean-tmp <0|1> (Default: 1)

If switched off, intermediate files won't be cleaned in the end.

--data-dir <dir> (Default: /usr/share/pdf2htmlEX)

Specify the folder holding the manifest and other files (see below for the manifest file)`

--css-draw <0|1> (Default: 0)

Experimental and unsupported CSS drawing

--debug <0|1> (Default: 0)

Print debug information.

Meta

-v, --version

Print copyright and version info

--help Print usage information

MANIFEST and DATA-DIR

When split-pages is 0, the manifest file describes how the final html page should be generated.

By default, pdf2htmlEX will use the manifest in the default data-dir (run `pdf2htmlEX -v` to check), which gives a simple demo of its syntax.

You can modify the default one, or you can create a new one and specify the correct data-dir in the command line.

When single-html is 1, all files referred by the manifest must be located in the data-dir.

EXAMPLE

pdf2htmlEX /path/to/file.pdf

Convert file.pdf into file.html

pdf2htmlEX --clean-tmp 0 --debug 1 /path/to/file.pdf

Convert file.pdf and leave all intermediate files.

pdf2htmlEX --dest-dir out --single-html 0 /path/to/file.pdf

Convert file.pdf into out/file.html and leave font/image files separated.

Last edited by Lu Wang, 2 months ago

# Customize Output

The output of pdf2htmlEX is flexible, you can tweak it as you like.

The files telling pdf2htmlEX how to compose the output are stored in data-dir. Run pdf2htmlEX -v to find the default location, or you may specify a new one with the parameter --data-dir.

data-dir/manifest is the template for the output. A simple syntax is supported, which is explained in the default manifest file. By default the following files are used in manifest

* base.css: Essential CSS styles that do not depend on specific PDF files
* fancy.css: A default UI theme
* jquery.css : A copy of jQuery, required by pdf2htmlEX.js
* pdf2htmlEX.js : A simple UI implementation for demonstration purpose

You may modify fancy.css to change the UI appearance. You can even include your own CSS files after the $css line, in order to override styles defined in PDF!

Note that the syntax of manifest is experimental, which might be changed in the future.

Last edited by coolwanglu, 4 months ago