

marqius

REPLACE WKHTMLTOPDF

THE SWITCHING MANUAL

Keep your code. Change the engine.

PYTHON · PHP · RUBY · .NET · NODE · GO · DOCKER

MARQIUS 0.11 · NOVEMBER 2026

Replace wkhtmltopdf: the switching manual

For teams whose apps make invoices, statements, reports and letters with wkhtmltopdf, through pdfkit, Snappy, laravel-snappy, wicked_pdf, Rotativa or any other wrapper. This manual shows how to swap the engine underneath in an afternoon, without touching your templates or your code, and how to prove the result on your own documents before anything goes live.

It covers one job only: getting off wkhtmltopdf safely. Every command in it can be copied straight from the page. All of them are also in **commands.txt**, a file attached to this PDF (look for the paperclip in your PDF viewer).

CONTENTS

1 Switch in sixty seconds 4
2 What you get 6
3 Install, update, undo 7
4 Why move now 11
5 Python: pdfkit, django-wkhtmltopdf, imgkit 12
6 PHP and Laravel: Snappy, laravel-snappy 14
7 Ruby and Rails: wicked_pdf, PDFKit 16
8 .NET: Rotativa, Wkhtmltopdf.NetCore, DinkToPdf 17
9 Node.js, Go and everything else 18
10 Docker and servers 19
11 Prove it on your own documents 22
12 What is different 25
13 Security and hardening 26
14 When you want more 27
15 Troubleshooting 28
16 Price, licence and support 29

1

Switch in sixty seconds



One line installs marqius-wkhtmltopdf and puts it where your wrapper library looks for wkhtmltopdf:

TERMINAL

```
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash
```

bash — a Linux server

```
$ curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash

  marqius-wkhtmltopdf · the maintained drop-in for wkhtmltopdf

  ✓ Platform     linux arm64
  ✓ Version      0.11.0 (latest)
  ✓ Downloaded   marqius-wkhtmltopdf-linux-arm64, 19 MB, checksum verified
  ✓ Installed    /usr/local/bin/marqius-wkhtmltopdf
  ✓ wkhtmltopdf  /usr/local/bin/wkhtmltopdf and wkhtmltoimage run marqius-wkhtmltopdf

  marqius-wkhtmltopdf 0.11.0 is installed. Programs that run wkhtmltopdf now run it.

  Next steps
    1. Check the version your libraries will see
       wkhtmltopdf -V
    2. Make a PDF the way your app does
       wkhtmltopdf --page-size A4 page.html page.pdf
    3. Compare it with a PDF wkhtmltopdf made before
       marqius-wkhtmltopdf compare old.pdf page.pdf

  Update   run the same command again
  Undo     run it again, ending  bash -s -- --uninstall
  Guide    https://marqius.com/download/marqius-wkhtmltopdf/replace-wkhtmltopdf.pdf
  Help     hello@marqius.com
```

Then check that your library will find it, and make a PDF exactly as you always did:

TERMINAL

```
wkhtmltopdf -V
wkhtmltopdf --page-size A4 invoice.html invoice.pdf
```

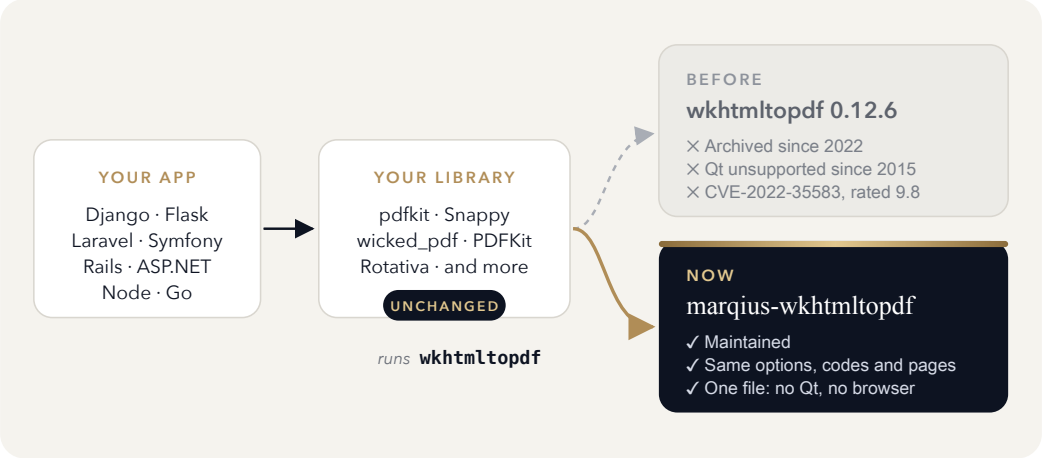


That is the switch. Your code, your templates and your wrapper library stay as they were. To update later, run the same line again; to undo it, see chapter 3.

2

What you get

marqius-wkhtmltopdf is a maintained program that answers to the name **wkhtmltopdf**. Your wrapper library calls it exactly as it called wkhtmltopdf, and gets back what it expects. **What it costs:** an afternoon to switch. Try it free for 30 days; then $10 a month, or $500 once for a lifetime licence.



WHAT STAYS THE SAME	WHAT IT MEANS FOR YOU
The program's name, `wkhtmltopdf` (and `wkhtmltoimage`)	Your library finds it where it found the old one
Every option of wkhtmltopdf 0.12.6, long and short	No option to rename, no template to touch
The page objects: `cover`, `toc`, pages, `-` for standard input and output	Covers, contents pages and piped output work as before
Exit codes, warnings and the closing "Done" line	Your error handling, logs and alerts keep working
The pages: shrink-to-fit, fonts, line breaks, page breaks	Customers see the documents they always saw

WHAT GETS BETTER	WHY IT MATTERS
It is maintained	Fixes ship; no security finding left open for good
It is fast	About 20 times faster with wkhtmltopdf's default settings; 4 to 8 times with its wait for scripts switched off
It is one program	No Qt, no X server, no browser, no old system libraries to hunt for
It updates in one line	Run the installer again; the version before is kept for a rollback
It has a way forward	Modern CSS, EU e-invoices, accessible PDFs: one option each, when you want them

3

Install, update, undo

The one-line installer

TERMINAL

```
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash
```

It finds your platform (Linux on x86-64 or ARM, or a Mac with Apple silicon), downloads the latest marqius-wkhtmltopdf, checks the download against its published checksum, and installs it in `/usr/local/bin`. It asks for your password through `sudo` only if that folder needs it; without `sudo`, it installs in `~/.local/bin`. Then it gives marqius-wkhtmltopdf the names wkhtmltopdf and wkhtmltoimage beside it.

Options

Options go after `bash -s --`, like this:

TERMINAL

```
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
```

OPTION	WHAT IT DOES
`--take-over`	A wkhtmltopdf is there already: take its place. The old program is kept, for `--uninstall`
`--no-wkhtmltopdf`	Install marqius-wkhtmltopdf only, without the names wkhtmltopdf and wkhtmltoimage
`--prefix DIR`	Install into another folder
`--version X.Y.Z`	Install a given version instead of the latest
`--rollback`	Go back to the version installed before the last update
`--uninstall`	Take marqius-wkhtmltopdf and the names away, and put back a wkhtmltopdf it replaced

When a wkhtmltopdf is already installed

The installer never replaces another program without being asked. It installs marqius-wkhtmltopdf, leaves the old wkhtmltopdf where it is, and tells you how to take its place. Run it again with `--take-over` when you are ready: the old program is kept beside marqius-wkhtmltopdf, so `--uninstall` can put it back.

bash — a Linux server

```
$ curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over

  marqius-wkhtmltopdf · the maintained drop-in for wkhtmltopdf

  ✓ Platform     linux arm64
  ✓ Version      0.11.0 (latest)
  ✓ Downloaded   marqius-wkhtmltopdf-linux-arm64, 19 MB, checksum verified
  ✓ Installed    /usr/local/bin/marqius-wkhtmltopdf
  ✓ wkhtmltopdf  /usr/local/bin/wkhtmltopdf taken over (the old one is kept)

  marqius-wkhtmltopdf 0.11.0 is installed. Programs that run wkhtmltopdf now run it.

  Next steps
    1. Check the version your libraries will see
       wkhtmltopdf -V
    2. Make a PDF the way your app does
       wkhtmltopdf --page-size A4 page.html page.pdf
    3. Compare it with a PDF wkhtmltopdf made before
       marqius-wkhtmltopdf compare old.pdf page.pdf

  Update   run the same command again
  Undo     run it again, ending  bash -s -- --uninstall
  Guide    https://marqius.com/download/marqius-wkhtmltopdf/replace-wkhtmltopdf.pdf
  Help     hello@marqius.com
$ wkhtmltopdf -V
wkhtmltopdf 0.12.6 (compatible; marqius-wkhtmltopdf 0.11.0)
$ ls /usr/local/bin
marqius-wkhtmltopdf
wkhtmltoimage
wkhtmltopdf
wkhtmltopdf.before-marqius
```

Update

Run the same line again. If a newer version is out, it is installed and the version before it is kept; if not, nothing changes.

TERMINAL

```
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash
```

Roll back, or take it all away

TERMINAL

```
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --rollback
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --uninstall
```

bash — a Linux server

```
$ curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --uninstall

  marqius-wkhtmltopdf · the maintained drop-in for wkhtmltopdf

  ✓ wkhtmltopdf  names taken away; a wkhtmltopdf it replaced is back
  ✓ Removed      /usr/local/bin/marqius-wkhtmltopdf
$ wkhtmltopdf -V
wkhtmltopdf 0.12.6 (with patched qt)
$ ls /usr/local/bin
wkhtmltopdf
```

If your team does not run scripts from the web

Download the script, read it, then run it:

TERMINAL

```
curl -fsSLO https://marqius.com/marqius-wkhtmltopdf.sh
less marqius-wkhtmltopdf.sh
bash marqius-wkhtmltopdf.sh
```

Or install by hand, checking the file against its checksum (on a Mac, use `shasum -a 256 -c` in place of `sha256sum -c`):

TERMINAL

```
v=0.12.0; f=marqius-wkhtmltopdf-linux-x86_64
curl -fsSLO https://marqius.com/download/marqius-wkhtmltopdf/$v/$f
curl -fsSLO https://marqius.com/download/marqius-wkhtmltopdf/$v/$f.sha256
sha256sum -c $f.sha256
chmod +x $f && sudo mv $f /usr/local/bin/marqius-wkhtmltopdf
sudo marqius-wkhtmltopdf install
```

Fonts

marqius-wkhtmltopdf uses the fonts installed on the server, as wkhtmltopdf did. On a server with no fonts at all, it stops with "no fonts are installed here" and prints the line that installs them on that very system. Here are the same lines. Each installs, in one go, the Liberation fonts, which have the letter widths of Arial, Times New Roman and Courier New, so lines break where they always did, and DejaVu Sans.

On Debian and Ubuntu:

TERMINAL

```
sudo apt-get update && sudo apt-get install -y fonts-liberation2 fonts-dejavu-core
```

On Alpine:

TERMINAL

```
apk add --no-cache font-liberation font-dejavu
```

On Fedora, Red Hat, Rocky, Alma and Amazon Linux 2023:

TERMINAL

```
sudo dnf install -y liberation-fonts dejavu-sans-fonts
```

Every other language. For Chinese, Japanese, Korean, Hindi, Thai and the rest of the world's writing systems, add the Noto fonts. On Debian and Ubuntu, about 130 MB:

TERMINAL

```
sudo apt-get install -y fonts-noto-core fonts-noto-cjk
```

On Alpine and Fedora, each writing system has a package of its own. These lines add Chinese, Japanese, Korean, Hindi and Thai:

TERMINAL

```
apk add --no-cache font-noto-cjk font-noto-devanagari font-noto-thai
```

TERMINAL

```
sudo dnf install -y google-noto-sans-cjk-fonts \
google-noto-sans-devanagari-fonts google-noto-sans-thai-fonts
```

marqius-wkhtmltopdf sets each letter in an installed font that has it, wherever that font is installed. When a page has letters no installed font has, it says so and prints the line that installs the fonts for those very letters, on that system.

Your own fonts. For company fonts, or any `.ttf`, `.otf` or `.ttc` file, put the files in a folder named `fonts`, then:

TERMINAL

```
sudo mkdir -p /usr/local/share/fonts && sudo cp fonts/* /usr/local/share/fonts/
```

There is no font cache to refresh: the next conversion uses them. On a Mac, double-click a font and choose Install; on Windows, right-click it and choose Install.

> Windows: the Windows build is in testing. On a Windows machine today, use WSL, which is Linux, and the installer works as above.

4

Why move now

wkhtmltopdf is finished

- Its code was archived after a last change in November 2022. The last release, 0.12.6, came out in 2020; the 0.12.7 its maintainer promised never came.
- By its own status page, the Qt it is built on hasn't been supported since 2015, and the WebKit browser engine inside it hasn't been updated since 2012.
- Searches for "wkhtmltopdf" fell by almost half in the twelve months to October 2026. Teams are moving off it now.

It is an open security finding

- **CVE-2022-35583**, a server-side request forgery in 0.12.6, is rated 9.8 out of 10 (critical) by the US National Vulnerability Database. HTML that reaches the converter can make it fetch internal addresses. There is no fixed version, and there won't be.
- The project's status page tells users not to run it on untrusted HTML, because that can lead to a takeover of the server.
- Scanners that check software against that database flag it, and auditors ask about it. Each audit means an exception to explain, every year.

The usual way out is a rewrite

Most teams leaving wkhtmltopdf move to a headless browser. For invoices and reports that means a full browser on every server, a new set of options, and pages that break in different places, so every template is checked and adjusted again. marqius-wkhtmltopdf is the other way out: keep the templates, keep the options, keep the pages, and change one program.

5

Python: pdfkit, django-wkhtmltopdf, imgkit

pdfkit (4.9 million downloads a month) doesn't make PDFs itself: it runs the wkhtmltopdf program and hands it your HTML. So does django-wkhtmltopdf. imgkit runs wkhtmltoimage. marqius-wkhtmltopdf answers to both names.

pdfkit

After the installer has run, there is nothing to change:

PYTHON

```python
import pdfkit

options = {"page-size": "A4", "margin-top": "15mm"}
pdfkit.from_file("invoice.html", "invoice.pdf", options=options)
```

If your code names the program's path, name marqius-wkhtmltopdf instead:

PYTHON

```python
config = pdfkit.configuration(wkhtmltopdf="/usr/local/bin/wkhtmltopdf")
pdfkit.from_string(html, "invoice.pdf", configuration=config)
```

pdfkit judges each run by wkhtmltopdf's exit code and its closing "Done" line. marqius-wkhtmltopdf writes them the same way, so a page that loads with a missing stylesheet is handled as it always was.

> When pdfkit says "No wkhtmltopdf executable found", it couldn't find the program. Run the installer, or pass `configuration=` as above.

django-wkhtmltopdf

Your views (`PDFTemplateView`, `PDFTemplateResponse`) stay as they are. Only if the program isn't on the PATH, add one line to `settings.py`:

PYTHON

```python
WKHTMLTOPDF_CMD = "/usr/local/bin/wkhtmltopdf"
```

imgkit

imgkit makes pictures of pages with wkhtmltoimage. marqius-wkhtmltopdf makes PNG, JPEG, BMP and SVG pictures, with the same width, height, crop and quality options.

PYTHON

```
config = imgkit.config(wkhtmltoimage="/usr/local/bin/wkhtmltoimage")
imgkit.from_file("chart.html", "chart.png", config=config)
```

In a Docker image

DOCKERFILE

```
FROM marqius/wkhtmltopdf:latest AS marqius
FROM python:3.12-slim
COPY --from=marqius /marqius-wkhtmltopdf /usr/local/bin/wkhtmltopdf
RUN ln -s /usr/local/bin/wkhtmltopdf /usr/local/bin/wkhtmltoimage
RUN apt-get update && apt-get install -y fonts-liberation2 fonts-dejavu-core
```

The same holds in Flask, FastAPI, Celery jobs and plain scripts: pdfkit is the same everywhere, and so is marqius-wkhtmltopdf under it.

6

PHP and Laravel: Snappy, laravel-snappy

KnpLabs Snappy (1.9 million installs a month) and barryvdh/laravel-snappy hand your HTML to the wkhtmltopdf program. Point them at marqius-wkhtmltopdf and nothing else in the app changes.

Laravel

The published `config/snappy.php` reads the program's path from your environment. In `.env`:

.ENV

```
WKHTML_PDF_BINARY=/usr/local/bin/wkhtmltopdf
WKHTML_IMG_BINARY=/usr/local/bin/wkhtmltoimage
```

Your controllers and Blade views don't change:

PHP

```
return PDF::loadView('invoices.show', $data)->setPaper('a4')->download('invoice.pdf');
```

If your app used the package h4cc/wkhtmltopdf-amd64, which ships the archived program inside `vendor/`, take it out and set the two lines above:

TERMINAL

```
composer remove h4cc/wkhtmltopdf-amd64
```

Symfony and plain PHP

PHP

```
use Knp\Snappy\Pdf;

$snappy = new Pdf('/usr/local/bin/wkhtmltopdf');
$snappy->generateFromHtml($html, '/tmp/invoice.pdf', ['page-size' => 'A4']);
```

With KnpSnappyBundle, set `binary` under `knp_snappy.pdf` to the same path.

> When Snappy says "The exit status code '1' says something went wrong", the program reported a failure. marqius-wkhtmltopdf fails with the same codes and messages for the same causes: read the next line to see which file couldn't be loaded.

Headers and footers

`--header-html`, `--footer-html`, `--header-center "[page] of [topage]"` and the other header and footer options behave as in wkhtmltopdf, page numbers and section names included.

7

Ruby and Rails: wicked_pdf, PDFKit

wicked_pdf and the PDFKit gem render your views by running wkhtmltopdf. Install marqius-wkhtmltopdf under that name, and your `render pdf:` calls, layouts and helpers stay exactly as they are.

wicked_pdf

1. **Drop the bundled program.** If your Gemfile has `gem "wkhtmltopdf-binary"`, remove that line, then run `bundle install`. That gem carries the archived wkhtmltopdf.
2. **Put marqius-wkhtmltopdf in place** with the installer, or tell wicked_pdf where it is:

```ruby
# config/initializers/wicked_pdf.rb
WickedPdf.config = { exe_path: "/usr/local/bin/wkhtmltopdf" }
```

3. **Restart the app.** Nothing else changes:

```ruby
render pdf: "invoice", template: "invoices/show", layout: "pdf"
```

wicked_pdf first asks the program for its version, then passes your pages as `file://` addresses, with the contents options after the word `toc`. marqius-wkhtmltopdf answers all of it as wkhtmltopdf does. The helpers `wicked_pdf_stylesheet_link_tag` and `wicked_pdf_image_tag` change your HTML, not the program, so they keep working.

> When wicked_pdf says "Bad wkhtmltopdf's path", `exe_path` points at nothing. Point it at marqius-wkhtmltopdf as above.

PDFKit (the Ruby gem)

```ruby
PDFKit.configure { |config| config.wkhtmltopdf = "/usr/local/bin/wkhtmltopdf" }
```

The gem's defaults, Letter paper, 0.75 inch margins and UTF-8, are passed on and come out the same.

8

.NET: Rotativa, Wkhtmltopdf.NetCore, DinkToPdf

Rotativa.AspNetCore

Rotativa runs wkhtmltopdf from a folder in your app, usually `wwwroot/Rotativa`.

1. Replace the `wkhtmltopdf` file in that folder with marqius-wkhtmltopdf, under the same name:

```
TERMINAL
cp /usr/local/bin/marqius-wkhtmltopdf wwwroot/Rotativa/wkhtmltopdf
chmod +x wwwroot/Rotativa/wkhtmltopdf
```

2. Nothing else changes:

```
C#
return new ViewAsPdf("Invoice", model) { PageSize = Size.A4 };
```

Rotativa sends the page on standard input and reads the PDF from standard output, with margins in millimetres and accented letters written as numbered character references. marqius-wkhtmltopdf handles all of it as wkhtmltopdf did.

Wkhtmltopdf.NetCore

```
C#
builder.Services.AddWkhtmltopdf("wkhtmltopdf");
```

Put marqius-wkhtmltopdf in the folder that line names, as `wkhtmltopdf`. Your `IGeneratePdf` calls stay as they are.

DinkToPdf and WkHtmlToPdf-DotNet

These two load wkhtmltopdf's library file, libwkhtmltox, into your process instead of running the program, so a program can't stand in for them. A marqius-wkhtmltopdf package with the same calls is planned: write to hello@marqius.com with "DinkToPdf" in the subject to be told first.

> ASP.NET Core apps in Linux containers can switch today. The Windows build of marqius-wkhtmltopdf is in testing.

9

Node.js, Go and everything else

If your code starts the wkhtmltopdf program, through a package or by hand, marqius-wkhtmltopdf takes its place: the same arguments, the same standard input and output, the same exit codes.

Node.js: the wkhtmltopdf package

JAVASCRIPT

```
const wkhtmltopdf = require("wkhtmltopdf");
wkhtmltopdf.command = "/usr/local/bin/wkhtmltopdf"; // only if it isn't on the PATH
wkhtmltopdf(html, { pageSize: "A4" }).pipe(fs.createWriteStream("report.pdf"));
```

Go: go-wkhtmltopdf

GO

```
wkhtmltopdf.SetPath("/usr/local/bin/wkhtmltopdf") // or set WKHTMLTOPDF_PATH
pdfg, _ := wkhtmltopdf.NewPDFGenerator()
pdfg.AddPage(wkhtmltopdf.NewPageReader(strings.NewReader(html)))
_ = pdfg.Create()
```

Java, Elixir, shell scripts and the rest

The command line is wkhtmltopdf's: global options first, then the documents (`cover`, `toc` and pages), each with its own options, and `-` for standard input or output. `--read-args-from-stdin` makes one PDF for each line it reads. Unknown or misplaced options fail with wkhtmltopdf's own messages, and with exit code 1.

10

Docker and servers

Add marqius-wkhtmltopdf to your own image

marqius-wkhtmltopdf is a single static program. Copy it into any Linux image under the name wkhtmltopdf:

DOCKERFILE

```
FROM marqius/wkhtmltopdf:latest AS marqius
COPY --from=marqius /marqius-wkhtmltopdf /usr/local/bin/wkhtmltopdf
RUN ln -s /usr/local/bin/wkhtmltopdf /usr/local/bin/wkhtmltoimage
```

Then give it fonts. The image's own take one line:

DOCKERFILE

```
COPY --from=marqius /usr/share/fonts /usr/share/fonts
```

Or install them with your system's packages: the lines in chapter 3, after `RUN`.

Switching from another wkhtmltopdf image

If your Dockerfile copies `/bin/wkhtmltopdf` out of surnet/alpine-wkhtmltopdf or icalialabs/wkhtmltopdf, change only the image's name. marqius/wkhtmltopdf has the program at the same path, and at `/bin/wkhtmltoimage`:

DOCKERFILE

```
COPY --from=marqius/wkhtmltopdf /bin/wkhtmltopdf /bin/wkhtmltopdf
```

Then delete the lines that install wkhtmltopdf's libraries (libx11, libxrender, libxext, libssl, fontconfig, freetype): marqius-wkhtmltopdf is one static program and needs none of them.

Run it as a command

TERMINAL

```
docker run --rm -v "$PWD:/data" -w /data marqius/wkhtmltopdf in.html out.pdf
```

Or with no folder to mount at all, through standard input and output:

TERMINAL

```
docker run --rm -i marqius/wkhtmltopdf - - < in.html > out.pdf
```

The image is free for 30 days, like the program.

Run it as a web service

One line starts wkhtmltopdf as a service: your app posts the HTML and gets the PDF back.

TERMINAL

```
docker run -d -p 8080:80 marqius/wkhtmltopdf serve
```

TERMINAL

```
curl -F file=@invoice.html localhost:8080 -o invoice.pdf
curl --data-binary @invoice.html 'localhost:8080/?page-size=A4' -o invoice.pdf
```

It answers the same requests as the wkhtmltopdf-as-a-service images (openlabs/docker-wkhtmltopdf-aas and the images built on it): a form with the page in `file` and wkhtmltopdf's options as JSON in `options`, or JSON with the page in base64 under `contents`. If you use one of those images, change only the image's name.

Add the invoice's data, and the PDF comes back as a Factur-X / ZUGFeRD e-invoice:

TERMINAL

```
curl -F file=@invoice.html -F einvoice=@invoice.xml localhost:8080 -o invoice.pdf
```

It is built to face your own apps safely. Each conversion runs on its own, with each option taken as a single value, so nothing in a request can run a command. Options that would read or write files on the server are refused, and local files stay blocked. A request's size (`MARQIUS_MAX_BODY`, 64 MB) and time (`MARQIUS_TIMEOUT`, 60 seconds) are limited. Keep it on your private network: like any converter that fetches web addresses, it should not face the open internet. Outside Docker, `marqius-wkhtmltopdf serve 8080` starts it on this machine only.

Fonts in the images

Every tag converts as it is:

TAG	FONTS IN IT
`latest`, `alpine`, `trixie`, `bookworm`, `bullseye`	Liberation Sans, Serif and Mono, and DejaVu Sans
`slim`	Liberation Sans alone, for the smallest download

To add a Linux server's fonts in one go, install them there (the lines are in chapter 3), then start the container with them. The image keeps its own fonts and gains the server's:

TERMINAL

```
docker run --rm -v /usr/share/fonts:/usr/local/share/fonts:ro \
-v "$PWD:/data" -w /data marqius/wkhtmltopdf in.html out.pdf
```

The web service takes them the same way:

TERMINAL

```
docker run -d -p 8080:80 -v /usr/share/fonts:/usr/local/share/fonts:ro \
marqius/wkhtmltopdf serve
```

A folder of your own fonts works the same way, on a Mac too:

TERMINAL

```
docker run --rm -v "$PWD/fonts:/usr/local/share/fonts:ro" \
-v "$PWD:/data" -w /data marqius/wkhtmltopdf in.html out.pdf
```

To build more fonts into your own image, start from a tag with a system. On Debian:

DOCKERFILE

```
FROM marqius/wkhtmltopdf:bookworm
RUN apt-get update && apt-get install -y --no-install-recommends fonts-noto-core \
fonts-noto-cjk && rm -rf /var/lib/apt/lists/*
COPY fonts/ /usr/local/share/fonts/
```

On Alpine:

DOCKERFILE

```
FROM marqius/wkhtmltopdf:alpine
RUN apk add --no-cache font-noto-cjk font-noto-devanagari font-noto-thai
COPY fonts/ /usr/local/share/fonts/
```

The `COPY` line adds a `fonts` folder of your own; leave it out if you have none.

11

Prove it on your own documents

A switch like this is signed off on evidence, not on a vendor's word. Here is how to gather that evidence in an hour.

A one-hour pilot

1. **Pick twenty documents** that matter: your busiest invoice, a long statement that runs over several pages, a report with a header and footer, something with a table of contents, and anything that ever caused trouble.
2. **Keep the PDFs wkhtmltopdf makes today** in a folder called `old`.
3. **Make the same documents with marqius-wkhtmltopdf**, with the same options, into a folder called `new`.
4. **Compare them**, one pair or the whole folders:

TERMINAL

```
marqius-wkhtmltopdf compare old/invoice.pdf new/invoice.pdf
marqius-wkhtmltopdf compare old/ new/ -o report/
```

5. **Read what it reports**, sign off, and switch.

bash

```
$ marqius-wkhtmltopdf compare old/invoice.pdf new/invoice.pdf
old/invoice.pdf  against  new/invoice.pdf
verdict: appearance only (review)
pages 1 and 1, words 99 and 99

how it looks: 1 difference
  moved           page 1   the line "Due: 17 Dec 2026" moved 1 pt to the left

the file itself: 1 difference
  pdf-type                the file is PDF 1.7, was PDF 1.4  [version]

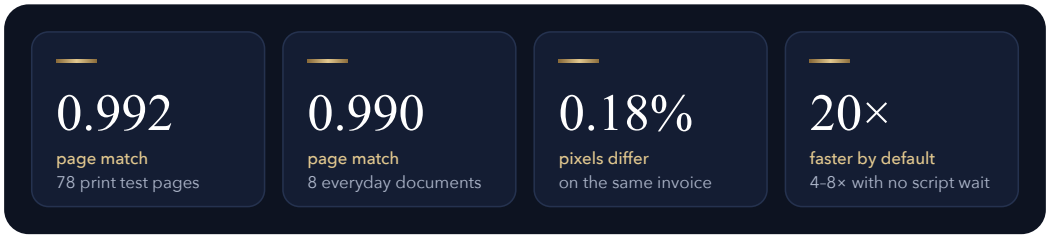
note: the old file was made by wkhtmltopdf 0.12.6 / Qt 4.8.7; the new one by marqius /
marqius 0.11.0 (not counted as a difference)
```

The report puts what a reader would notice first: words that changed, lines that moved, pages that broke elsewhere, pictures that went missing. Then come the details of the file itself, such as the PDF version, which no reader sees.

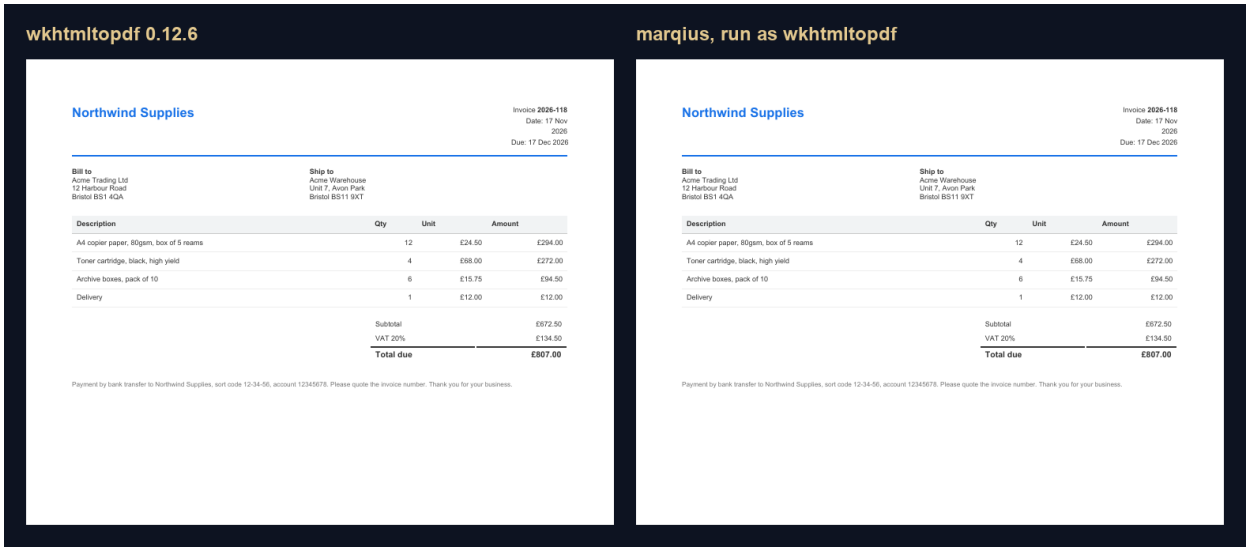
Two folders are paired by file name, and their report goes in the folder after `-o`: a summary of every pair, and for each document that differs, both PDFs marked where they differ and pictures of those pages.

What we measured

We ran 86 test documents through wkhtmltopdf 0.12.6 and through marqius-wkhtmltopdf, in the same Linux container, with the same fonts. Each was scored on its text (half the score), on where every word lands on the page (three tenths), and on its page count (one fifth).



TEST SET	DOCUMENTS	SCORE OUT OF 1
Print test pages: text, tables, lists, headers, page breaks, contents	78	0.992
Everyday documents: invoice, receipt, statement, letter, label, report, certificate, columns	8	0.990



On this invoice only 0.18% of the pixels differ. Even the date that wraps onto a second line in wkhtmltopdf wraps the same way in marqius-wkhtmltopdf. The two lowest scores belong to test pages built

for unusual corners: rare writing systems, where the fonts on the machine decide, and see-through elements (chapter 12).

How fast

The same documents, run one after another through each program on the same Linux machine, with nothing else running:

TEST SET	WKHTMLTOPDF	MARQIUS-WKHTMLTOPDF	FASTER BY
78 print test pages, default settings	19.45 s	0.91 s	21.4 times
8 everyday documents, default settings	2.07 s	0.10 s	20.7 times
78 print test pages, `--javascript-delay 0`	3.40 s	0.44 s	7.7 times
8 everyday documents, `--javascript-delay 0`	0.31 s	0.07 s	4.4 times

By default wkhtmltopdf waits 200 ms on every page for scripts to finish, which is most of its time; libraries such as pdfkit and Snappy leave that wait as it is. The last two rows switch it off for both programs, to show the difference that remains.

12

What is different

marqius-wkhtmltopdf aims to make the same pages as wkhtmltopdf, so most differences come from places where wkhtmltopdf itself was unusual. Here is the full list we know of.

WHAT	WKHTMLTOPDF	MARQIUS-WKHTMLTOPDF
Pages that build themselves with JavaScript	Runs the scripts in its 2012 browser engine	Runs them in Chrome or Chromium when one is installed; without one, runs scripts that write into the page as it loads
See-through (opacity) parts of a page	Turns them into pictures; their text can't be selected	Keeps them as real, searchable text
An HTML header file without `<!DOCTYPE>`	The header disappears	The header is drawn
`--xsl-style-sheet` for the contents	Applies the stylesheet	Draws its own contents page, and warns that the stylesheet was not applied
A background picture cut to the content box	Drawn faded	Drawn as it is

What wkhtmltopdf's browser engine never understood (flexbox, grid, calc(), CSS variables), marqius-wkhtmltopdf passes over too, so your pages keep their old layout. Add `--marqius-modern-css` when you want those to work.

13

Security and hardening

Five steps for any HTML converter

1. **Keep local file access off.** It is off by default, in wkhtmltopdf 0.12.6 and in marqius-wkhtmltopdf. Open single folders with `--allow /path/to/assets`, never everything.
2. **Escape user input** before it reaches your HTML templates.
3. **Turn scripts off** if your documents don't need them: `--disable-javascript`.
4. **Fence the converter in.** Run it in a container with no route to internal services: databases, admin panels, cloud metadata addresses.
5. **Keep it up to date.** With marqius-wkhtmltopdf, that is the installer's one line; with wkhtmltopdf it is no longer possible.

What marqius-wkhtmltopdf changes

- It is maintained, so fixes ship when they are needed.
- It is written in Rust. No 2012 browser engine runs inside your process.
- It keeps the same safe defaults and the same switches as 0.12.6, so your existing hardening carries over.

Like any converter that fetches web addresses, marqius-wkhtmltopdf should still be fenced in, as in step 4, when it handles HTML you don't control.

14

When you want more

You don't need any of this to switch. Each item is one option on the same command line, for the day you want it.

YOU WANT	ADD THIS OPTION	WHAT YOU GET
Modern CSS	`--marqius-modern-css`	Flexbox, grid, calc() and custom properties are laid out, instead of passed over
An EU e-invoice inside the PDF	`--marqius-einvoice invoice.xml`	A Factur-X / ZUGFeRD invoice: your PDF with the invoice data built in, as PDF/A-3
The e-invoice's profile, named by you	`--marqius-einvoice-profile en16931`	The profile the PDF declares (minimum, basicwl, basic, en16931, extended or xrechnung), instead of the one read from the invoice
Files carried inside the PDF	`--marqius-attach order.xml`	The file travels inside the PDF, such as the order or a UBL e-invoice; repeat the option for more files
An accessible PDF	`--marqius-ua`	A tagged PDF (PDF/UA) that screen readers can follow
An archival PDF	`--marqius-pdfa 2b`	A PDF/A file for long-term records; the level is 2a, 2b, 2u, 3a, 3b, 3u, 4 or 4f
Your details in the file	`--marqius-author`, `--marqius-subject`, `--marqius-keywords`, `--marqius-lang`	The document's properties, as a records system reads them

15 Troubleshooting

WHAT YOU SEE	WHY	WHAT TO DO
"No wkhtmltopdf executable found" (pdfkit)	pdfkit can't find the program	Run the installer, or pass the path to pdfkit
"Bad wkhtmltopdf's path" (wicked_pdf)	`exe_path` points at nothing	Set `exe_path` to `/usr/local/bin/wkhtmltopdf`
"The exit status code '1' says something went wrong" (Snappy)	Something on the page couldn't be loaded	Read the next line: it names the file or address
"Warning: Blocked access to file …", then exit code 1	Local file access is off by default, as in 0.12.6	Add `--enable-local-file-access`, or `--allow` with the folder
"Exit with code 1 due to network error" after "Done"	The PDF was made, but something on the page didn't load	Wrappers treat this as before; fix the address, or ignore it as you did
"Host … not found" for a file name	The file isn't where the command says; like wkhtmltopdf, marqius-wkhtmltopdf then takes the name for a web address	Check the path
"Error: no fonts are installed here"	The server, or the image, has no fonts at all	Run the line printed under it. Chapter 3 has the same lines; chapter 10 has the Docker ways
Letters look different from before	A font your page names isn't installed	Install it, or Liberation fonts for Arial, Times New Roman and Courier New
"Warning: no font has a glyph for …", and boxes in the PDF	No font on the server has those letters (Chinese or Hindi, say)	Run the line printed under it, or see Fonts in chapter 3
A chart drawn by JavaScript is missing	No Chrome or Chromium on the server	Install Chromium, or draw the chart on the server as an SVG
`wkhtmltopdf -V` doesn't mention marqius-wkhtmltopdf	Your PATH finds another wkhtmltopdf first	Run `which -a wkhtmltopdf`, then rerun the installer with `--take-over`

Exit codes are wkhtmltopdf's: 0 when all went well; 1 when a document couldn't be read or made, or when something on a page couldn't be loaded. In that last case the PDF is still written, followed by "Done" and the reason.

16

Price, licence and support

- **Try it free for 30 days; then $10 a month, or $500 once for a lifetime licence.** Buy at marqius.com/buy.
- **One licence covers one app**, on as many servers and containers as it runs on: production, staging and development alike.
- **The lifetime licence includes every future version.** Prices are subject to change; see marqius.com for the latest prices.
- marqius-wkhtmltopdf is proprietary software from an independent publisher. It is not affiliated with or endorsed by the wkhtmltopdf project.
- **Support:** write to hello@marqius.com. Say which wrapper library you use and which version of it, and attach the HTML that gave the trouble if you can.

The whole command line

YOU TYPE	IT DOES
`curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash`	Installs marqius-wkhtmltopdf, or updates it
`wkhtmltopdf -V`	Prints `wkhtmltopdf 0.12.6 (compatible; marqius-wkhtmltopdf 0.12.0)`
`wkhtmltopdf -h`	The common options
`wkhtmltopdf -H`	Every option of 0.12.6
`wkhtmltopdf [options] [cover …] [toc …] page … out.pdf`	Makes the PDF, as wkhtmltopdf would
`marqius-wkhtmltopdf …`	The same, without installing the name
`marqius-wkhtmltopdf install [folder] [--force]`	Installs the names wkhtmltopdf and wkhtmltoimage
`marqius-wkhtmltopdf install --undo`	Puts back the program that was there before
`marqius-wkhtmltopdf compare old.pdf new.pdf`	Lists every difference between two PDFs
`marqius-wkhtmltopdf compare old/ new/ -o report/`	The same for two folders, with the report in `report/`
`marqius-wkhtmltopdf serve [port]`	wkhtmltopdf as a web service: post the HTML, get the PDF
`marqius-wkhtmltopdf --version`	Prints `marqius-wkhtmltopdf 0.12.0`