

marqius

REPLACE WKHTMLTOPDF

THE DEVELOPER MANUAL

Your library stays. The engine changes.

PYTHON · PHP · RUBY · NODE · GO · JAVA · ELIXIR · .NET

MARQIUS-WKHTMLTOPDF · NOVEMBER 2026

marqius-wkhtmltopdf with your library

For developers who switch a project from wkhtmltopdf to marqius-wkhtmltopdf. Fifteen libraries run wkhtmltopdf for the apps that use them, in Python, PHP, Ruby, Node, Go, Java, Elixir and .NET. Each has a chapter here: how to point it at marqius-wkhtmltopdf, the command lines it sends, how to test the switch, and what is known to differ.

Every command in this manual was run as printed on a fresh Linux machine, and all of them are attached to this PDF as `commands.txt`.

CONTENTS

1 Before you start 3
2 python-pdfkit 7
3 imgkit 10
4 django-wkhtmltopdf 13
5 KnpLabs Snappy 16
6 laravel-snappy 18
7 phpwkhtmltopdf 20
8 KnpSnappyBundle 23
9 CakePdf 26
10 wicked_pdf 29
11 PDFKit (Ruby) 31
12 wkhtmltopdf for Node 33
13 go-wkhtmltopdf 35
14 java-wkhtmltopdf-wrapper 38
15 pdf_generator (Elixir) 41
16 Rotativa.AspNetCore 44
17 Help and licence 47

1 Before you start

marqius-wkhtmltopdf is a maintained program that takes the place of wkhtmltopdf 0.12.6. It reads the same command lines, writes to the same places and ends with the same exit statuses, so the library in your app never knows the program changed. It answers to the names `wkhtmltopdf` and `wkhtmltoimage`, and needs no X server, no Qt and no system libraries.

Install it

On a Linux server, or a Mac with Apple silicon, one line installs it in `/usr/local/bin` (or `~/.local/bin` for a user without administrator rights) and gives it the names `wkhtmltopdf` and `wkhtmltoimage`:

TERMINAL

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

Where wkhtmltopdf is installed already, the installer leaves it in place until you say so. This line puts marqius-wkhtmltopdf in its place and keeps the old program, so that `--uninstall` can put it back:

TERMINAL

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

In a Docker image, three lines after your `FROM` line copy in the program and its fonts, and give it the two names:

DOCKERFILE

```
FROM debian:bookworm-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

Delete the lines that installed wkhtmltopdf and its libraries (libx11, libxrender, libxext, libssl, fontconfig, xvfb): marqius-wkhtmltopdf needs none of them.

Help in the program

From the release after 0.12.5, the program prints any library's setting with the paths where marqius-wkhtmltopdf is installed on that server, and says whether those paths run it:

TERMINAL

```
marqius-wkhtmltopdf help laravel-snappy
```

`marqius-wkhtmltopdf help libraries` lists the fifteen libraries and the names it takes for each; a language, such as `php`, gives all of its libraries. Each chapter names its own.

How each chapter is laid out

- **The library:** what it is and how many people use it.
- **Switching:** installing, and the library's own setting for the program's path, if you need it.
- **The command lines it sends:** the real ones, recorded while the library's own test suite ran.
- **Testing the switch:** a smoke test made with the library before and after the switch, the two PDFs compared, then the library's own test suite with marqius-wkhtmltopdf in place.
- **Known differences and limits.**

The commands were checked as root in the language's official Docker image on Debian 12 (python, php, ruby, node, golang, elixir) or in plain Debian 12 (Java, .NET), on a server where wkhtmltopdf 0.12.6.1 ran before the switch. On your own machine, add `sudo` where your system needs it, and work in your project's usual environment, such as a Python virtual environment.

What we measured

On 19 November 2026 we ran each library's own test suite on one machine with wkhtmltopdf 0.12.6.1, upstream's last build, and again with marqius-wkhtmltopdf in its place: 0.12.4, then a build with the fixes that are now 0.12.5. A shim in front of the program kept every call each suite made: those are the command lines quoted in each chapter. Only the tests that pass with wkhtmltopdf count:

LIBRARY	PASS WITH WKHTMLTOPDF	ALSO WITH 0.12.4	ALSO WITH 0.12.5'S FIXES
python-pdfkit	47 of 47	47	47
imgkit	45 of 48	44	45
django-wkhtmltopdf	14 of 18	14	14
KnpLabs Snappy	49 of 49	49	49
laravel-snappy	9 of 9	9	9
phpwkhtmltopdf	34 of 34	34	34
KnpSnappyBundle	26 of 26	26	26
CakePdf	32 of 34	32	32
wicked_pdf	61 of 61	58	61
PDFKit (Ruby)	180 of 182	180	180
wkhtmltopdf for Node	5 of 5	5	5
go-wkhtmltopdf	34 of 35	34	34
java-wkhtmltopdf-wrapper	30 of 30	29	30
pdf_generator (Elixir)	12 of 15	12	12
Rotativa.AspNetCore	10 of 10	10	10

The tests of KnpLabs Snappy, KnpSnappyBundle and CakePdf check those libraries' own code and never run the program; their chapters' smoke tests prove the switch. The five tests that pass with wkhtmltopdf and not with 0.12.4 come from three faults, all fixed in 0.12.5:

- **Waiting for the window status.** 0.12.4 did not wait for `--window-status`. 0.12.5 waits, as wkhtmltopdf does, until the page's scripts set the text (java-wkhtmltopdf-wrapper).
- **Links that are not stylesheets.** 0.12.4 fetched a page's `dns-prefetch`, `preconnect`, `icon` and `manifest` links as files. Those named bare servers, which answered "not found", so making a picture of github.com ended with status 1. 0.12.5 fetches only stylesheets, as a browser does (imgkit).
- **The PDF's first line.** 0.12.4's files started with `%PDF-1.7`. wkhtmltopdf's start with `%PDF-1.4`, and wicked_pdf's tests check for it. 0.12.5 writes `%PDF-1.4` as wkhtmltopdf does (wicked_pdf).

Compare the PDFs

`compare` reads two PDFs the way a person does and names every difference by its kind:

TERMINAL

```
marqius-wkhtmltopdf compare old.pdf new.pdf
```

The verdict is `same`, `appearance only` (the same words, with something moved or drawn a little differently) or `content differs` (words, numbers or pictures changed). It ends with status 0 when nothing differs and 1 when something does; in a test pipeline, `--fail-on blocking` ends with 1 only when words, numbers, dates or pictures changed. For the smoke tests in this manual, made with marqius-wkhtmltopdf 0.12.4, it answered the same every time:

RECORDED

```
old.pdf  against  new.pdf
verdict: appearance only (review)
pages 1 and 1, words 6 and 6

how it looks: 3 differences
  moved           page 1   2 lines from "Invoice 1001 Total due: 120.00 EUR": the words
                           stand up to 7 pt from where they stood along the line
  font            page 1   2 words set in liberationsans bold are now in
                           liberationserif bold
  font            page 1   4 words set in liberationsans are now in liberationserif

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 0.12.6.1 / Qt 4.8.7; the new one by
marqius / marqius 0.12.4 (not counted as a difference)
```

The same words on the same pages, with two differences known:

- **A page that names no font.** The smoke test's page has no styles, and wkhtmltopdf sets such text in its sans-serif font (Liberation Sans on these machines), where 0.12.4 sets it in the serif one; the words then stand a little apart. A page that names its fonts, as templates usually do, gets the same fonts from both.
- **The PDF version in the file's first line.** From 0.12.5 it reads 1.4, as wkhtmltopdf's does.

For many documents at once, put the old PDFs in one folder and the new ones, with the same names, in another:

TERMINAL

```
marqius-wkhtmltopdf compare old/ new/ -o report/
```

2
python-pdfkit

The library

python-pdfkit (`pip install pdfkit`) is the most used way to make PDFs from Python: 5.1 million downloads a month. It runs the wkhtmltopdf program, hands it your HTML on standard input and reads the PDF back. ERPNext and Frappe print every invoice through it.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM python:3.13-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
RUN pip install --no-cache-dir pdfkit
```

pdfkit runs the first `wkhtmltopdf` on the PATH, as `which wkhtmltopdf` finds it, so most apps change nothing. If your code names the program's path, name `/usr/local/bin/wkhtmltopdf`, as the smoke test below does:

PYTHON · your code

```
config = pdfkit.configuration(wkhtmltopdf="/usr/local/bin/wkhtmltopdf")
```

From the release after 0.12.5, `marqius-wkhtmltopdf help python-pdfkit` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf --quiet - -
wkhtmltopdf --page-size Letter --quiet - out.pdf
wkhtmltopdf --outline --footer-line --quiet --quiet - -
wkhtmltopdf --page-size Letter --cookie test_cookie1 "" \
    --cookie test_cookie2 cookie_value2 --quiet - -
```

Your options come first, then `--quiet` (unless you pass `verbose=True`), then `-`: the HTML comes on standard input. The PDF goes to the file you name, or to standard output (`-`) when you ask for it as bytes. With `--quiet`, pdfkit raises an error on any exit status other than 0. marqius-wkhtmltopdf ends with the same status as wkhtmltopdf for the same page, so pdfkit raises exactly where it raised before.

Testing the switch

A smoke test with pdfkit's own call, saved as `smoke.py`:

PYTHON · smoke.py

```
import sys

import pdfkit

config = pdfkit.configuration(wkhtmltopdf="/usr/local/bin/wkhtmltopdf")
html = "<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>"
pdfkit.from_string(html, sys.argv[1], configuration=config)
```

Make the PDF with wkhtmltopdf, switch, make it again with marqius-wkhtmltopdf, and compare the two:

TERMINAL

```
pip install pdfkit
python3 smoke.py old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
python3 smoke.py new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then pdfkit's own test suite, with marqius-wkhtmltopdf in place:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/JazzCore/python-pdfkit.git
cd python-pdfkit/tests
pip install pytest
python3 -m pytest pdfkit-tests.py
```

All 47 tests pass.

Known differences and limits

- None found by its test suite.
- pdfkit reads options from `<meta name="pdfkit-...">` tags in your HTML and passes them on as options: they reach marqius-wkhtmltopdf the same way.
- With `verbose=True`, pdfkit leaves out `--quiet` and prints the program's progress lines; marqius-wkhtmltopdf prints the same lines.

3
imgkit

The library

imgkit (`pip install imgkit`) makes pictures of web pages with wkhtmltoimage: 454,000 downloads a month. marqius-wkhtmltopdf answers to that name too, and makes PNG, JPEG, BMP and SVG pictures with the same width, height, crop and quality options.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM python:3.13-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
RUN pip install --no-cache-dir imgkit
```

imgkit runs the first `wkhtmltoimage` on the PATH. If your code names the program's path, name `/usr/local/bin/wkhtmltoimage`:

PYTHON · your code

```
config = imgkit.config(wkhtmltoimage="/usr/local/bin/wkhtmltoimage")
```

If your code passes `xvfb=` to `imgkit.config` or `xvfb=True` to the calls, take it out: marqius-wkhtmltopdf needs no X server.

From the release after 0.12.5, `marqius-wkhtmltopdf help imgkit` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltoimage - -
wkhtmltoimage --format jpg - out.jpg
wkhtmltoimage fixtures/example.html out.jpg
wkhtmltoimage https://github.com out.jpg
```

The page comes on standard input (`-`), or as a file or an address; the picture goes to the file you name, or to standard output. imgkit raises an error when the program ends with any status other than 0.

Testing the switch

A smoke test with imgkit's own call, saved as `smoke.py`:

PYTHON · smoke.py

```
import sys

import imgkit

config = imgkit.config(wkhtmltoimage="/usr/local/bin/wkhtmltoimage")
html = "<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>"
imgkit.from_string(html, sys.argv[1], config=config)
```

Make the picture with wkhtmltoimage, switch, and make it again:

TERMINAL

```
pip install imgkit
python3 smoke.py old.png
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
python3 smoke.py new.png
```

`compare` reads PDFs, not pictures: open `old.png` and `new.png` side by side. To compare the page itself, make it as a PDF with `wkhtmltopdf` before and after, and compare those PDFs as chapter 1 shows.

Then imgkit's own test suite. Some of its tests run the program under xvfb-run, so install that too:

TERMINAL

```
apt-get update
apt-get install -y --no-install-recommends xvfb xauth
cd "$(mktemp -d)"
git clone https://github.com/jarrekk/imgkit.git
cd imgkit/tests
pip install pytest six
python3 -m pytest imgkit_test.py
```

With marqius-wkhtmltopdf 0.12.4, 44 tests pass and 4 fail. Three of those fail with wkhtmltoimage too, on any machine like ours: `test_raise_error_if_bad_wkhtmltoimage_option` calls `assertRegexpMatches`, which Python 3.12 removed, and `test_cmd_not_found` and `test_no_where_which` need `sudo` and a system whose `/bin` is not `/usr/bin`. The fourth, `test_from_url`, makes a picture of github.com: it passes with 0.12.5.

Known differences and limits

- 0.12.4 fetched a page's `dns-prefetch`, `preconnect`, `icon` and `manifest` links as if they were files. Where those name bare servers, as github.com's do, the picture is made but the program ends with status 1, and imgkit raises an error. 0.12.5 fetches only stylesheets.

4
django-wkhtmltopdf

The library

django-wkhtmltopdf gives Django views that return PDFs (`PDFTemplateView`, `PDFTemplateResponse`): 103,000 downloads a month. It renders your template to a temporary file, runs wkhtmltopdf on it and reads the PDF from standard output.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM python:3.13-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
RUN pip install --no-cache-dir django django-wkhtmltopdf
```

django-wkhtmltopdf runs `WKHTMLTOPDF_CMD` from your settings, then from the environment, and otherwise the first `wkhtmltopdf` on the PATH. To name the program, add this line to your project's `settings.py`:

PYTHON · add to mysite/settings.py

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

From the release after 0.12.5, `marqius-wkhtmltopdf help django-wkhtmltopdf` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf --encoding utf8 --quiet /tmp/wkhtmltopdfb7009mgl.html -
wkhtmltopdf --encoding utf8 --quiet --title ♥ /tmp/tmpqnw3_5fe.html -
wkhtmltopdf --encoding utf8 --footer-html /tmp/wkhtmltopdf0jn6qdgm.html --quiet \
    /tmp/wkhtmltopdfb3zsz6t5.html -
```

Options are sorted by name and all come before the page, header and footer files included. The page is a temporary file; the PDF comes back on standard output. Any exit status other than 0 raises `CalledProcessError`.

Testing the switch

A smoke test that uses your project's settings and django-wkhtmltopdf's own call, saved as `smoke.py` beside `manage.py` (with your project's name in place of `mysite`):

PYTHON · smoke.py

```
import os
import sys

import django

os.environ.setdefault("DJANGO_SETTINGS_MODULE", "mysite.settings")
django.setup()

from wkhtmltopdf.utils import wkhtmltopdf

with open("invoice.html", "w") as f:
    f.write("<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>")
with open(sys.argv[1], "wb") as f:
    f.write(wkhtmltopdf(["invoice.html"]))
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
python3 smoke.py old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
python3 smoke.py new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then django-wkhtmltopdf's own test suite. It runs on Django 4.0, the newest its own tests support, which needs Python 3.10 or older:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/incuna/django-wkhtmltopdf.git
cd django-wkhtmltopdf
pip install "Django>=4.0,<4.1" six -r test_requirements.txt
python3 wkhtmltopdf/tests/run.py
```

14 of its 18 tests pass, as with wkhtmltopdf. The other 4 fail with wkhtmltopdf 0.12.6.1 too: their footer loads a script and a picture as `file://` addresses, which 0.12.6 blocks unless it is given `--enable-local-file-access`, so the program ends with status 1 (`Exit with code 1 due to network error: ProtocolUnknownError`).

Known differences and limits

- None found by its test suite.
- django-wkhtmltopdf turns your `STATIC_URL` and `MEDIA_URL` addresses into `file://` addresses. wkhtmltopdf 0.12.6 and marqius-wkhtmltopdf both block those by default, as the 4 tests above show. If your templates use them, set `WKHTMLTOPDF_CMD_OPTIONS = {"quiet": True, "enable_local_file_access": True}` in your settings.
- django-wkhtmltopdf's list of options without a value still names `--verbose`, which wkhtmltopdf 0.12.6 does not have. Both programs refuse it in the same way.

5
KnpLabs Snappy

The library

Snappy (`composer require knplabs/knp-snappy`) is PHP's wrapper for wkhtmltopdf and wkhtmltoimage: 2.0 million installs a month. laravel-snappy and KnpSnappyBundle are built on it, and OpenEMR prints its patient reports through it.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM php:8.4-cli
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

Snappy runs the program you give its constructor, and checks that the file can be run. Name `/usr/local/bin/wkhtmltopdf`, and `/usr/local/bin/wkhtmltoimage` for pictures:

PHP · your code

```
$snappy = new Pdf('/usr/local/bin/wkhtmltopdf');
```

From the release after 0.12.5, `marqius-wkhtmltopdf help snappy` prints this setting with the paths on your server.

The command lines it sends

Snappy's own tests never run the program. This one is from laravel-snappy's tests, which run it through Snappy, and the second from the smoke test below:

RECORDED

```
wkhtmltopdf --lowquality /tmp/knp_snappy6afe6bcf3605a8.36596182.html \
    /tmp/knp_snappy6afe6bcf3609b5.88814737.pdf
wkhtmltopdf --lowquality /tmp/knp_snappy6afe7bc2e07939.47671103.html old.pdf
```

`--lowquality` is Snappy's only default option. Every page is a file (an HTML string is written to a temporary `knp_snappy*.html` first) and the PDF always goes to a file, never to standard output. Snappy runs the command through a shell and raises an error when the exit status is not 0 and standard error is not empty; it does not pass `--quiet`, so that holds for any failure. Then it checks that the PDF exists and is not empty.

Testing the switch

A smoke test with Snappy's own call, saved as `smoke.php`:

PHP · smoke.php

```php
<?php
require __DIR__ . '/vendor/autoload.php';

use Knp\Snappy\Pdf;

$snappy = new Pdf('/usr/local/bin/wkhtmltopdf');
$html = '<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>';
$snappy->generateFromHtml($html, $argv[1]);
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
composer require knplabs/knp-snappy
php smoke.php old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
php smoke.php new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then Snappy's own test suite:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/KnpLabs/snappy.git
cd snappy
composer update --prefer-dist --no-progress
vendor/bin/phpunit
```

All 49 tests pass. They test Snappy's own code (its commands and options) without running the program, so the smoke test above and laravel-snappy's suite are what prove the switch.

Known differences and limits

- With Snappy's `--lowquality`, marqius-wkhtmltopdf stores pictures at no more than 150 dots per inch and JPEG quality 75, for a smaller file. Text and lines stay sharp.
- Snappy's list of options still names options of wkhtmltopdf 0.9 (`toc-depth`, `toc-font-name` and others). If your code sets one, both programs refuse it in the same way.

6
laravel-snappy

The library

laravel-snappy (`composer require barryvdh/laravel-snappy`) brings Snappy to Laravel, with the `PDF` and `SnappyImage` facades: 872,000 installs a month.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM php:8.4-fpm
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

laravel-snappy reads the program's path from `WKHTML_PDF_BINARY` and `WKHTML_IMG_BINARY`, and its defaults are `/usr/local/bin/wkhtmltopdf` and `/usr/local/bin/wkhtmltoimage`: where the installer puts marqius-wkhtmltopdf. If your `.env` names another path, name these:

.ENV · add to .env

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

If your app caches its configuration, run `php artisan config:clear` after the change. If it used the package h4cc/wkhtmltopdf-amd64, which carries the old program inside `vendor/`, take it out with `composer remove h4cc/wkhtmltopdf-amd64`.

From the release after 0.12.5, `marqius-wkhtmltopdf help laravel-snappy` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf --lowquality /tmp/knp_snappy6afe6bcf3605a8.36596182.html \
    /tmp/knp_snappy6afe6bcf3609b5.88814737.pdf
wkhtmltopdf --lowquality \
    /tmp/snappy-test-6afe6bd0bd759/knp_snappy6afe6bd0bd7986.42517274.html \
    /tmp/snappy-test-6afe6bd0bd759/test.pdf
```

They are Snappy's command lines (chapter 5), with your `options` from `config/snappy.php`; `setPaper()` adds `--page-size` and `--orientation`.

Testing the switch

In your Laravel app, a smoke test with the `PDF` facade: make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
php artisan tinker --execute="PDF::loadHTML('<h1>Invoice 1001</h1>')->save('old.pdf')"
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
php artisan tinker --execute="PDF::loadHTML('<h1>Invoice 1001</h1>')->save('new.pdf')"
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then laravel-snappy's own test suite:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/barryvdh/laravel-snappy.git
cd laravel-snappy
composer update --prefer-dist --no-progress
vendor/bin/phpunit
```

All 9 tests pass, and they run the program 7 times.

Known differences and limits

- None found by its test suite. Snappy's notes in chapter 5 hold here too.

7
phpwkhtmltopdf

The library

phpwkhtmltopdf (`composer require mikehaertl/phpwkhtmltopdf`) is a PHP wrapper with its own way of building commands: 715,000 installs a month. Drupal's Entity Print module uses it for its wkhtmltopdf engine.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM php:8.4-cli
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

phpwkhtmltopdf runs `wkhtmltopdf` and `wkhtmltoimage` from the PATH unless you give the `binary` option. To name the program:

PHP · your code

```
$pdf = new Pdf(['binary' => '/usr/local/bin/wkhtmltopdf']);
```

If your code sets `enableXvfb` in `commandOptions`, take it out: marqius-wkhtmltopdf needs no X server.

From the release after 0.12.5, `marqius-wkhtmltopdf help phpwkhtmltopdf` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf /tmp/php_tmpfile_9hrvsjtacpfb0U837PT.html \
    /tmp/tmp_wkhtmlto_pdf_0jo8do30mdak0RzzdnD.pdf
wkhtmltopdf cover /tmp/tmp_wkhtmlto_pdf_cnr9urbm8gpsbt2zeMg.html \
    /tmp/tmp_wkhtmlto_pdf_hucj26jjh2lbceAy11X.pdf
wkhtmltopdf --no-outline --margin-top 0 --header-center "test {x} {y}" \
    /root/dlt/run-wk/tests/assets/test.html --no-background --zoom 1.5 \
    --cookie name value --replace x v --replace y "" \
    cover /root/dlt/run-wk/tests/assets/test.html --replace x a --replace y b \
    toc --disable-dotted-lines /tmp/tmp_wkhtmlto_pdf_bmcc2fpt83r60OIkdsI.pdf
wkhtmltoimage --transparent --width 800 --allow /tmp --allow /test \
    /root/dlt/run-wk/tests/assets/test.html \
    /tmp/tmp_wkhtmlto_pdf_i8tkd3ed9ocd9drDl7k.png
```

HTML strings are written to temporary files; each page's options follow that page, and covers and contents pages are objects of their own. The PDF or picture always goes to a temporary file, which `saveAs()` then copies. A run fails when the exit status is not 0, unless `ignoreWarnings` is set and the file was still made.

Testing the switch

A smoke test with phpwkhtmltopdf's own call, saved as `smoke.php`:

PHP · smoke.php

```php
<?php
require __DIR__ . '/vendor/autoload.php';

use mikehaertl\wkhtmlto\Pdf;

$pdf = new Pdf(['binary' => '/usr/local/bin/wkhtmltopdf']);
$pdf->addPage('<html><h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p></html>');
if (!$pdf->saveAs($argv[1])) {
    exit($pdf->getError() . "\n");
}
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
composer require mikehaertl/phpwkhtmltopdf
php smoke.php old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
php smoke.php new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then phpwkhtmltopdf's own test suite. Some of its tests run the program under xvfb-run, so install that too:

TERMINAL

```
apt-get update
apt-get install -y --no-install-recommends xvfb xauth
cd "$(mktemp -d)"
git clone https://github.com/mikehaertl/phpwkhtmltopdf.git
cd phpwkhtmltopdf
composer update --prefer-dist --no-progress
vendor/bin/phpunit
```

All 34 tests pass, and they run the program 34 times.

Known differences and limits

- None found by its test suite.
- phpwkhtmltopdf treats a string as HTML only when it contains `<html`; anything else is taken for a file name or an address, as before.

8 KnpSnappyBundle

The library

KnpSnappyBundle (`composer require knplabs/knp-snappy-bundle`) brings Snappy to Symfony as the services `knp_snappy.pdf` and `knp_snappy.image`: 551,000 installs a month. Sylius's invoicing and refund plugins print through it.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM php:8.4-fpm
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

The bundle runs `binary` under `knp_snappy.pdf` and `knp_snappy.image`; without one, `wkhtmltopdf` and `wkhtmltoimage` from the PATH. To name the programs, in `config/packages/knp_snappy.yaml`:

YAML · config/packages/knp_snappy.yaml

```
knp_snappy:
    pdf:
        binary: /usr/local/bin/wkhtmltopdf
    image:
        binary: /usr/local/bin/wkhtmltoimage
```

If your file reads the paths from `.env` instead (`'%env(WKHTMLTOPDF_PATH)%'`), set `WKHTMLTOPDF_PATH` and `WKHTMLTOIMAGE_PATH` there to the same two paths.

From the release after 0.12.5, `marqius-wkhtmltopdf help knpsnappybundle` prints this setting with the paths on your server.

The command lines it sends

The bundle hands its options to Snappy, so it sends Snappy's command lines (chapter 5). Recorded from the smoke test below:

RECORDED

```
wkhtmltopdf --lowquality /tmp/knp_snappy6afe7bc5659849.54425196.html old.pdf
```

Testing the switch

In your Symfony app, a smoke test that boots the kernel and takes the bundle's service, saved as `smoke.php` beside `composer.json`:

PHP · smoke.php

```php
<?php
use App\Kernel;
use Knp\Snappy\Pdf;
use Symfony\Component\Dotenv\Dotenv;

require __DIR__ . '/vendor/autoload.php';

(new Dotenv())->bootEnv(__DIR__ . '/.env');
$kernel = new Kernel($_SERVER['APP_ENV'], (bool) $_SERVER['APP_DEBUG']);
$kernel->boot();
$pdf = $kernel->getContainer()->get(Pdf::class);
$pdf->generateFromHtml('<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>', $argv[1]);
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
php smoke.php old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
php smoke.php new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then the bundle's own test suite:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/KnpLabs/KnpSnappyBundle.git
cd KnpSnappyBundle
composer update --prefer-dist --no-progress
vendor/bin/phpunit
```

All 26 tests pass. Like Snappy's, they never run the program, so the smoke test is what proves the switch.

Known differences and limits

- None known beyond Snappy's (chapter 5).

9 CakePdf

The library

CakePdf (`composer require friendsofcake/cakepdf`) makes PDFs in CakePHP apps with one of several engines, wkhtmltopdf among them: 80,000 installs a month.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM php:8.4-fpm
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

CakePdf's WkHtmlToPdf engine runs `binary` from its configuration, by default `wkhtmltopdf` from the PATH. To name the program, in `config/bootstrap.php`:

PHP · your code

```
Configure::write('CakePdf', [
    'engine' => [
        'className' => 'CakePdf.WkHtmlToPdf',
        'binary' => '/usr/local/bin/wkhtmltopdf',
    ],
]);
```

From the release after 0.12.5, `marqius-wkhtmltopdf help cakepdf` prints this setting with the paths on your server.

The command lines it sends

CakePdf's own tests never run the program. Recorded from the smoke test below:

RECORDED

```
wkhtmltopdf --quiet --print-media-type --orientation portrait --page-size A4 \
    --encoding UTF-8 - -
```

The HTML goes in on standard input and the PDF comes back on standard output. CakePdf adds `--margin-*` in millimetres, `--title`, `--javascript-delay` and `--window-status` when you set them, and `--header-*` and `--footer-*` text. It raises an error only when nothing comes back on standard output.

Testing the switch

A smoke test with CakePdf's own call, saved as `smoke.php`:

PHP · smoke.php

```php
<?php
require __DIR__ . '/vendor/autoload.php';

use Cake\Core\Configure;
use CakePdf\Pdf\CakePdf;

Configure::write('CakePdf', [
    'engine' => [
        'className' => 'CakePdf.WkHtmlToPdf',
        'binary' => '/usr/local/bin/wkhtmltopdf',
    ],
]);
$pdf = new CakePdf();
$pdf->write($argv[1], true, '<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>');
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
composer require friendsofcake/cakepdf
php smoke.php old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
php smoke.php new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then CakePdf's own test suite. It needs the PHP extensions CakePHP uses; in the official php image, `docker-php-ext-install` builds them:

TERMINAL

```
apt-get update
apt-get install -y --no-install-recommends libicu-dev libzip-dev libpng-dev \
    libjpeg62-turbo-dev libfreetype6-dev
docker-php-ext-configure gd --with-freetype --with-jpeg
docker-php-ext-install intl gd zip bcmath
cd "$(mktemp -d)"
git clone https://github.com/FriendsOfCake/CakePdf.git
cd CakePdf
composer update --prefer-dist --no-progress
vendor/bin/phpunit
```

32 tests pass and 2 are skipped, as with wkhtmltopdf. Its tests check CakePdf's own code without running the program, so the smoke test is what proves the switch.

Known differences and limits

- None found.

10
wicked_pdf

The library

wicked_pdf (`gem "wicked_pdf"`) renders Rails views as PDFs with `render pdf:`: 88.6 million downloads in all, the most of any wkhtmltopdf library for Ruby.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM ruby:3.3-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

If your Gemfile has `gem "wkhtmltopdf-binary"`, remove it with `bundle remove wkhtmltopdf-binary`: that gem carries the old program, and wicked_pdf would find it first. wicked_pdf runs the `exe_path` you configure, and otherwise looks for `wkhtmltopdf` with Bundler, then on the PATH. To name the program, in `config/initializers/wicked_pdf.rb`:

RUBY · config/initializers/wicked_pdf.rb

```
WickedPdf.configure do |config|
  config.exe_path = "/usr/local/bin/wkhtmltopdf"
end
```

From the release after 0.12.5, `marqius-wkhtmltopdf help wicked_pdf` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf -V
wkhtmltopdf file:////tmp/wicked_pdf20261119-6382-d21xk2.html \
    /tmp/wicked_pdf_generated_file20261119-6382-2u9gcs.pdf
```

wicked_pdf first asks the program for its version and reads `wkhtmltopdf 0.12.6`, so it writes `toc` and `cover` as 0.12 expects. marqius-wkhtmltopdf answers `wkhtmltopdf 0.12.6 (with patched qt)`, as wkhtmltopdf's own build does, and then names itself in brackets. The page goes in as a `file://` address of a temporary file (with four slashes, as wicked_pdf writes it), and the PDF goes to a temporary file. wicked_pdf ignores the exit status; it raises an error only when the PDF is empty.

Testing the switch

In your Rails app, a smoke test with wicked_pdf's own call, saved as `smoke.rb`:

RUBY · smoke.rb

```ruby
html = "<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>"
File.binwrite(ARGV[0], WickedPdf.new.pdf_from_string(html))
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
bin/rails runner smoke.rb old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
bin/rails runner smoke.rb new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then wicked_pdf's own test suite. It runs in a Rails 7.0 app that its Rakefile generates; two pins keep that app working with today's gems (minitest 5.24, and the logger loaded first):

TERMINAL

```
apt-get update
apt-get install -y --no-install-recommends libsqlite3-dev
cd "$(mktemp -d)"
git clone https://github.com/mileszs/wicked_pdf.git
cd wicked_pdf
printf "\ngem 'minitest', '~> 5.24.0'\n" >> gemfiles/7.0.gemfile
export BUNDLE_GEMFILE=$PWD/gemfiles/7.0.gemfile RUBYOPT=-rlogger
bundle install
bundle exec rake dummy_generate
WKHTMLTOPDF_BIN=/usr/local/bin/wkhtmltopdf bundle exec rake test
```

With marqius-wkhtmltopdf 0.12.4, 58 of the 61 tests pass. The other 3 check that the PDF starts with `%PDF-1.4`, and pass with 0.12.5.

Known differences and limits

- 0.12.4 starts its PDFs with `%PDF-1.7`. wicked_pdf's own tests check for `%PDF-1.4`; your app does not. 0.12.5 starts them with `%PDF-1.4`, as wkhtmltopdf does.
- wicked_pdf can send options that wkhtmltopdf 0.12.6 no longer has (`--toc-depth`, `--toc-font-name` and the other old contents-page options). Both programs refuse them in the same way.

11
PDFKit (Ruby)

The library

PDFKit (`gem install pdfkit`) makes PDFs from HTML in Ruby, with a Rack middleware that serves any page as a PDF: 31.0 million downloads in all. Gumroad makes its receipts with it.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM ruby:3.3-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

PDFKit runs `config.wkhtmltopdf` when you set it, and otherwise looks for `wkhtmltopdf` with Bundler, then on the PATH. If your Gemfile has `gem "wkhtmltopdf-binary"`, remove it. To name the program:

RUBY · your code

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

From the release after 0.12.5, `marqius-wkhtmltopdf help pdfkit-ruby` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf --quiet --page-size Letter --margin-top 0.75in --margin-right 0.75in \
    --margin-bottom 0.75in --margin-left 0.75in --encoding UTF-8 - -
wkhtmltopdf --quiet --page-size Letter --margin-top 0.75in --margin-right 0.75in \
    --margin-bottom 0.75in --margin-left 0.75in --encoding UTF-8 \
    --javascript-delay 1234 - -
```

PDFKit's defaults come first: Letter paper, 0.75 inch margins and UTF-8. The HTML goes in on standard input and the PDF comes back on standard output. Any exit status other than 0 raises an error, except status 2 when you set `load-error-handling` to `skip` or `ignore`. An empty PDF raises one too.

Testing the switch

A smoke test with PDFKit's own call, saved as `smoke.rb`:

RUBY · smoke.rb

```
require "pdfkit"

PDFKit.configure { |config| config.wkhtmltopdf = "/usr/local/bin/wkhtmltopdf" }
kit = PDFKit.new("<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>")
kit.to_file(ARGV[0])
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
gem install pdfkit
ruby smoke.rb old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
ruby smoke.rb new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then PDFKit's own test suite. Its CI runs it with Rails 7.0, and today's gems need the logger loaded first:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/pdfkit/pdfkit.git
cd pdfkit
export RAILS_VERSION="~> 7.0.0" RUBYOPT=-rlogger
bundle install
bundle exec rspec
```

180 of its 182 tests pass and 2 are pending, as with wkhtmltopdf.

Known differences and limits

- None found by its test suite.
- Eight of the middleware's tests compare the size of two PDFs made from the same page, and their names include that size, so they read differently for each program. They pass with both.

12
wkhtmltopdf for Node

The library

The `wkhtmltopdf` package for Node (`npm install wkhtmltopdf`) streams PDFs from HTML or addresses: 301,000 downloads a month. Forward Email makes its payment receipts with it.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM node:22-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

The package runs `wkhtmltopdf.command`, by default `wkhtmltopdf` from the PATH. To name the program:

JAVASCRIPT · your code

```
wkhtmltopdf.command = "/usr/local/bin/wkhtmltopdf";
```

From the release after 0.12.5, `marqius-wkhtmltopdf help node` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf --quiet - -
wkhtmltopdf --quiet http://localhost:42396/validFile.html -
wkhtmltopdf --quiet file:////root/dlt/run-wk/spec/fixtures/validFile.html -
```

The package builds one command line and runs it through `/bin/bash` (`/bin/sh` on a Mac), with the PDF piped through `| cat`. HTML goes in on standard input; an `http`, `https` or `file` address is

passed as it is. Any exit status other than 0 is an error, unless your `ignore` option matches what the program printed.

Testing the switch

A smoke test with the package's own call, saved as `smoke.js`:

JAVASCRIPT · smoke.js

```
const fs = require("fs");
const wkhtmltopdf = require("wkhtmltopdf");

wkhtmltopdf.command = "/usr/local/bin/wkhtmltopdf";
wkhtmltopdf("<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>", { pageSize: "A4" })
  .pipe(fs.createWriteStream(process.argv[2]));
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
npm install wkhtmltopdf
node smoke.js old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
node smoke.js new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then the package's own test suite:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/devongovett/node-wkhtmltopdf.git
cd node-wkhtmltopdf
npm install
npm test
```

All 5 tests pass.

Known differences and limits

- None found by its test suite.

13
go-wkhtmltopdf

The library

go-wkhtmltopdf (`github.com/SebastiaanKlippert/go-wkhtmltopdf`) is the Go wrapper with the most users: 1,200 stars on GitHub. It builds wkhtmltopdf's whole command line, cover and contents pages included.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM golang:1.25
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

go-wkhtmltopdf runs the path you give `SetPath`; without it, a `wkhtmltopdf` in your program's own folder, then the first on the PATH, then the one in the folder named by `WKHTMLTOPDF_PATH`. To name the program:

GO · your code

```
wkhtmltopdf.SetPath("/usr/local/bin/wkhtmltopdf")
```

From the release after 0.12.5, `marqius-wkhtmltopdf help go-wkhtmltopdf` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf page - -
wkhtmltopdf --version -
wkhtmltopdf --margin-bottom 0.5cm --margin-left 2cm --margin-right 1mm \
    --margin-top 10mm --page-height 10in --page-width 5.5in \
    page https://www.google.com -
wkhtmltopdf toc --enable-local-file-access --xsl-style-sheet testdata/toc.xls \
    --footer-html testdata/footer-toc.html --footer-spacing 8.000 \
    page - --enable-local-file-access --footer-html testdata/footer.html \
    --footer-spacing 8.000 -
```

Every page is written after the word `page`, with its own options after it. A page read from Go comes in on standard input (`-`); the PDF goes to standard output and into a buffer, or to `OutputFile`. Any exit status other than 0 is an error that carries what the program printed.

Testing the switch

A smoke test with go-wkhtmltopdf's own calls, saved as `main.go`:

GO · main.go

```go
package main

import (
    "log"
    "os"
    "strings"

    wkhtmltopdf "github.com/SebastiaanKlippert/go-wkhtmltopdf"
)

func main() {
    wkhtmltopdf.SetPath("/usr/local/bin/wkhtmltopdf")
    pdfg, err := wkhtmltopdf.NewPDFGenerator()
    if err != nil {
        log.Fatal(err)
    }
    html := "<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>"
    pdfg.AddPage(wkhtmltopdf.NewPageReader(strings.NewReader(html)))
    if err := pdfg.Create(); err != nil {
        log.Fatal(err)
    }
    if err := pdfg.WriteFile(os.Args[1]); err != nil {
        log.Fatal(err)
    }
}
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
go mod init smoke
go get github.com/SebastiaanKlippert/go-wkhtmltopdf
go run . old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
go run . new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then go-wkhtmltopdf's own test suite:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/SebastiaanKlippert/go-wkhtmltopdf.git
cd go-wkhtmltopdf
go test ./...
```

34 tests pass and one is skipped, as with wkhtmltopdf. Its tests make PDFs of google.com and wkhtmltopdf.org, so the machine needs to reach the internet.

Known differences and limits

- `--xsl-style-sheet` for the contents page: marqius-wkhtmltopdf draws its own contents page and says on standard error that the stylesheet was not applied. The test that sends it passes with both programs.
- 0.12.4 did not wait for `--window-status`. go-wkhtmltopdf's test of it stops the program after 200 milliseconds, and 0.12.4 passed it only because it had not finished by then. 0.12.5 waits, as wkhtmltopdf does.

14
java-wkhtmltopdf-wrapper

The library

java-wkhtmltopdf-wrapper (`com.github.jhonnymertz:java-wkhtmltopdf-wrapper` on Maven Central) is the Java wrapper with the most users: 333 stars on GitHub. It builds pages, covers and contents pages from strings, files and addresses.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM eclipse-temurin:17-jre
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

The wrapper runs the command you give `WrapperConfig`; `WrapperConfig.findExecutable()` finds `wkhtmltopdf` with `which`. To name the program:

JAVA · your code

```
Pdf pdf = new Pdf(new WrapperConfig("/usr/local/bin/wkhtmltopdf"));
```

If your code sets an `XvfbConfig`, take it out: marqius-wkhtmltopdf needs no X server.

From the release after 0.12.5, `marqius-wkhtmltopdf help java` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran, with the temporary files' long names shortened:

RECORDED

```
wkhtmltopdf /tmp/java-wkhtmltopdf-wrapper23ed4fee-...6512204.html -
wkhtmltopdf --javascript-delay 2000 http://www.google.com -
wkhtmltopdf --window-status complete http://www.google.com -
wkhtmltopdf cover /tmp/java-wkhtmltopdf-wrapper6540a8b5-...8078401475.html \
    toc --toc-header-text TableOfContents \
    /tmp/java-wkhtmltopdf-wrappere6047e3b-...5456286167186.html \
    --exclude-from-outline -
```

Pages made from strings are written to temporary files; the PDF comes back on standard output. A run fails on any exit status other than 0, unless you allow that status with `setSuccessValues`.

Testing the switch

A smoke test with the wrapper's own calls. Save this as `pom.xml`:

XML · pom.xml

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0">
  <modelVersion>4.0.0</modelVersion>
  <groupId>com.example</groupId>
  <artifactId>smoke</artifactId>
  <version>1</version>
  <properties>
    <maven.compiler.source>17</maven.compiler.source>
    <maven.compiler.target>17</maven.compiler.target>
  </properties>
  <dependencies>
    <dependency>
      <groupId>com.github.jhonnymertz</groupId>
      <artifactId>java-wkhtmltopdf-wrapper</artifactId>
      <version>1.3.1-RELEASE</version>
    </dependency>
  </dependencies>
</project>
```

And this as `src/main/java/Smoke.java`:

JAVA · src/main/java/Smoke.java

```java
import com.github.jhonnymertz.wkhtmltopdf.wrapper.Pdf;
import com.github.jhonnymertz.wkhtmltopdf.wrapper.configurations.WrapperConfig;

public class Smoke {
    public static void main(String[] args) throws Exception {
        Pdf pdf = new Pdf(new WrapperConfig("/usr/local/bin/wkhtmltopdf"));
        pdf.addPageFromString("<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>");
        pdf.saveAs(args[0]);
    }
}
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
mvn -q compile
mvn -q exec:java -Dexec.mainClass=Smoke -Dexec.args=old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
mvn -q exec:java -Dexec.mainClass=Smoke -Dexec.args=new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then the wrapper's own test suite. Some of its tests run the program under xvfb-run, so install that too:

TERMINAL

```
apt-get update
apt-get install -y --no-install-recommends xvfb xauth
cd "$(mktemp -d)"
git clone https://github.com/jhonnymertz/java-wkhtmltopdf-wrapper.git
cd java-wkhtmltopdf-wrapper
mvn -B verify
```

With marqius-wkhtmltopdf 0.12.4, 29 of the 30 tests pass. The other, `testPdfWithWindowStatusTimeoutParameters`, expects the program to wait for `--window-status`, and passes with 0.12.5.

Known differences and limits

- 0.12.4 did not wait for `--window-status`: it made the PDF at once. 0.12.5 waits, as wkhtmltopdf does, until the page's scripts set the text, and keeps waiting if they never do, so use it with a time limit.

15
pdf_generator (Elixir)

The library

pdf_generator (`{:pdf_generator, "~> 0.6.2"}` on Hex) makes PDFs in Elixir and Phoenix apps with wkhtmltopdf, or with Chrome: 4.0 million downloads in all.

Switching

Install marqius-wkhtmltopdf on the server:

TERMINAL

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

Or in your Docker image:

DOCKERFILE

```
FROM elixir:1.16-slim
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /usr/local/bin/marqius-wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
RUN marqius-wkhtmltopdf install
```

pdf_generator runs `wkhtml_path` from your configuration, and otherwise the first `wkhtmltopdf` on the PATH. To name the program, in `config/config.exs`:

ELIXIR · your code

```
config :pdf_generator, wkhtml_path: "/usr/local/bin/wkhtmltopdf"
```

If your configuration sets `command_prefix` to `xvfb-run`, take it out: marqius-wkhtmltopdf needs no X server.

From the release after 0.12.5, `marqius-wkhtmltopdf help pdf_generator` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf --page-size A4 /tmp/06zn1hqT.html /tmp/06zn1hqT.pdf
wkhtmltopdf --orientation landscape --margin-right 0 --page-size A4 \
    /tmp/P49RxtZU.html /tmp/P49RxtZU.pdf
```

The HTML goes to a temporary file and the PDF to another. pdf_generator does not pass `--quiet`, and it judges a run by the word `Done` in what the program printed, not by its exit status. marqius-wkhtmltopdf prints `Done` as wkhtmltopdf does.

Testing the switch

A smoke test in a small Mix project. Save this as `mix.exs`:

ELIXIR · mix.exs

```
defmodule Smoke.MixProject do
  use Mix.Project

  def project do
    [app: :smoke, version: "0.1.0", deps: [{:pdf_generator, "~> 0.6.2"}]]
  end

  def application do
    [extra_applications: [:logger]]
  end
end
```

This as `config/config.exs`:

ELIXIR · config/config.exs

```
import Config

config :pdf_generator, wkhtml_path: "/usr/local/bin/wkhtmltopdf"
```

And this as `smoke.exs`, with pdf_generator's own call:

ELIXIR · smoke.exs

```
[out] = System.argv()
html = "<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>"
File.write!(out, PdfGenerator.generate_binary!(html))
```

Make the PDF with wkhtmltopdf, switch, make it again, and compare:

TERMINAL

```
mix local.hex --force
mix deps.get
mix run smoke.exs old.pdf
curl -fsSL https://marqius.com/marqius-wkhtmltopdf.sh | bash -s -- --take-over
mix run smoke.exs new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Then pdf_generator's own test suite. It runs the program under xvfb-run, so install that too:

TERMINAL

```
apt-get update
apt-get install -y --no-install-recommends xvfb xauth
cd "$(mktemp -d)"
git clone https://github.com/gutschilla/elixir-pdf-generator.git
cd elixir-pdf-generator
mix local.rebar --force
mix deps.get
mix test
```

12 of its 15 tests pass, as with wkhtmltopdf. The 3 others are its Chrome tests, which need Chrome's Node package.

Known differences and limits

- None found by its test suite.

16
Rotativa.AspNetCore

The library

Rotativa.AspNetCore (`dotnet add package Rotativa.AspNetCore`) turns ASP.NET Core views into PDFs with `ViewAsPdf` and into pictures with `ViewAsImage`: 2.8 million downloads in all.

Switching

Rotativa runs `wkhtmltopdf` and `wkhtmltoimage` from a folder of your app, `wwwroot/Rotativa` unless your `RotativaConfiguration.Setup` names another. Put marqius-wkhtmltopdf there under those names. On the server, install it:

TERMINAL

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

Before you replace the program in your app's folder, make a PDF with it, exactly as Rotativa calls it, to compare later. From your project's folder:

TERMINAL

```
echo '<h1>Invoice 1001</h1><p>Total due: 120.00 EUR</p>' > invoice.html
wwwroot/Rotativa/wkhtmltopdf -q - - < invoice.html > old.pdf
```

Then put marqius-wkhtmltopdf in its place:

TERMINAL

```
cp /usr/local/bin/marqius-wkhtmltopdf wwwroot/Rotativa/wkhtmltopdf
cp /usr/local/bin/marqius-wkhtmltopdf wwwroot/Rotativa/wkhtmltoimage
```

The second copy is needed only for `ViewAsImage`. In the final stage of your Docker image, after the published app is copied to `/app`:

DOCKERFILE

```
FROM mcr.microsoft.com/dotnet/aspnet:10.0
WORKDIR /app
COPY --from=marqius/wkhtmltopdf /marqius-wkhtmltopdf /app/wwwroot/Rotativa/wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
```

Your `app.UseRotativa()` and your views stay as they are.

From the release after 0.12.5, `marqius-wkhtmltopdf help rotativa` prints this setting with the paths on your server.

The command lines it sends

Recorded while its own test suite ran:

RECORDED

```
wkhtmltopdf -q - -
wkhtmltoimage -q - -
wkhtmltoimage -q -f png --quality 90 - -
```

Rotativa sends the HTML on standard input, with every letter outside ASCII written as a numbered character reference, and reads the PDF or picture from standard output. Margins are plain numbers, in millimetres. It never looks at the exit status: it raises an error only when nothing comes back.

Testing the switch

Make the same PDF again, now with marqius-wkhtmltopdf, and compare the two:

TERMINAL

```
wwwroot/Rotativa/wkhtmltopdf -q - - < invoice.html > new.pdf
marqius-wkhtmltopdf compare old.pdf new.pdf
```

`compare` answers `appearance only`, with the differences chapter 1 explains.

Saving a PDF from your app's own pages before and after the switch, and comparing those, proves the same for your views.

Rotativa's own test suite runs in Docker: it builds its demo app, starts it and checks every kind of PDF and picture through Chromium. Its image installs wkhtmltopdf; one more image puts marqius-wkhtmltopdf in its place:

TERMINAL

```
cd "$(mktemp -d)"
git clone https://github.com/webgio/Rotativa.AspNetCore.git
cd Rotativa.AspNetCore
docker build -f Rotativa.AspNetCore.Tests/Dockerfile -t rotativa-tests .
docker build -t rotativa-tests-marqius - <<'EOF'
FROM rotativa-tests
COPY --from=marqius/wkhtmltopdf /bin/wkhtmltopdf /usr/local/bin/wkhtmltopdf
COPY --from=marqius/wkhtmltopdf /bin/wkhtmltoimage /usr/local/bin/wkhtmltoimage
COPY --from=marqius/wkhtmltopdf /usr/share/fonts /usr/share/fonts
EOF
docker run --rm rotativa-tests-marqius
```

All 10 tests pass, as with wkhtmltopdf.

Known differences and limits

- None found by its test suite.

- On Windows, Rotativa runs `wkhtmltopdf.exe` from the same folder. These steps were checked on Linux servers and in Linux containers.

17
Help and licence

marqius-wkhtmltopdf is free for 30 days on each machine. Then it is $10 a month, or $500 once for a lifetime licence: marqius.com/licence/marqius-wkhtmltopdf. After the trial, without a licence, it keeps making every PDF and adds one small line at the foot of each page.

- **The switching manual**, with Docker, fonts, security and every option: marqius.com/download/replace-wkhtmltopdf.pdf
- **Runnable examples for all 14 PDF libraries**: github.com/marqius-wkhtmltopdf/marqius-wkhtmltopdf-examples. Each example is checked in public CI on every commit.
- **Help**: write to hello@marqius.com with the library, its version, the command line from its chapter and what the program printed.