pdfly

Release 0.3.0

pdfly contributors

Jan 17, 2024

USER GUIDE

pdfly is a command line tool to get informaiton about PDF documents and to manipulate them.

INSTALLATION

There are several ways to install pdfly. The most common option is to use pip.

1.1 pip

pdfly requires Python 3.6+ to run.

Typically Python comes with `pip`, a package installer. Using it you can install pypdf:

```
pip install pdfly
```

If you are not a super-user (a system administrator / root), you can also just install pypdf for your current user:

```
pip install --user pdfly
```

1.2 pipx

We recommend to install pdfly via pipx:

```
pipx install pdfly
```

pipx installs the pdfly application in an isolated environment. That guarantees that no other applications interferes with its defpendencies.

1.3 Python Version Support

If ✓ is givien, it works. It is tested via CI. If ✕ is given, it is guaranteed not to work. If it's not filled, we don't guarantee support, but it might still work.

Python	3.12	3.11	3.10	3.9	3.8	3.7	3.6	2.7
pdfly	✓	✓	✓	✓	✓			✕

1.4 Development Version

In case you want to use the current version under development:

```
pip install git+https://github.com/py-pdf/pdfly.git
```

META

Get metadata of a PDF file.

2.1 Usage

```
pdfly meta --help

 Usage: pdfly meta [OPTIONS] PDF

 Show metadata of a PDF file

─ Arguments ─────────────────────────────────────────────────────────
│ *    pdf      FILE  [default: None] [required]                       │
──────────────────────────────────────────────────────────────────────
─ Options ───────────────────────────────────────────────────────────
│ --output  -o      [json|text]  output format [default: text]        │
│ --help                         Show this message and exit.          │
──────────────────────────────────────────────────────────────────────
```

2.2 Example

```
$pdfly meta Allianz-Versicherungsunterlagen.pdf

                       Operating System Data

         Attribute  Value

         File Name │ /home/user/Documents/Allianz-Versicherungsunterlagen.pdf
   File Permissions │ -rw-rw-r--
         File Size │ 874,781 bytes
     Creation Time │ 2023-09-02 10:00:51
 Modification Time │ 2023-09-02 10:00:42
       Access Time │ 2023-09-09 11:57:41

                           PDF Data

         Attribute  Value
```

```
              Title
           Producer   itext-paulo-155 (itextpdf.sf.net-lowagie.com)
             Author
              Pages   34
          Encrypted   None
   PDF File Version    %PDF-1.6
        Page Layout
          Page Mode
             PDF ID   ID1=b"'\xc5\x92\xc3\x92\xe2\x80\x93--/\xef\xac\x824\xc3...
                       ID2=b'\xc3\x8b\xc3\xaa\xcb\x9b\r\xc3\xa2\r\xcb\x99T\xc3...
                       \xc3\x96\xc3\x9fY2'
 Fonts (unembedded)    /Helvetica
   Fonts (embedded)    /ASPNQQ+TT22D6t00, /CBKSHX+Helvetica-Bold,
                       /CXQKAY+Helvetica, /GOCSXU+AllianzNeo-Bold,
                       /LKNHUL+Arial-BoldMT, /LMNFKX+ArialMT, /MWUNIP+Symbol,
                       /ODNMDG+TT5B6t00, /PESMKN+AllianzNeo-CondensedBold,
                       /PHDALA+Helvetica-Oblique, /PJEFXS+AllianzNeo-Light,
                       /SNDABN+Helvetica, /SNDABN+Helvetica-Bold,
                       /SNDABN+Times-Roman, /TXDAYK+Helvetica,
                       /VORXLN+Helvetica-BoldOblique, /YTXZAH+Arial-ItalicMT
        Attachments    []
             Images    16 images (355,454 bytes)
```

Use the 'pagemeta' subcommand to get details about a single page

CAT

The cat command can split / extract pages from a PDF. It can also join/merge/combine multiple PDF documents into a single one.

3.1 Usage

```
pdfly cat --help

 Usage: pdfly cat [OPTIONS] FILENAME FN_PGRGS...

 Concatenate pages from PDF files into a single PDF file.
 Page ranges refer to the previously-named file. A file not followed by a page
 range means all the pages of the file.
 PAGE RANGES are like Python slices.
 Remember, page indices start with zero.
 Page range expression examples:

    :      all pages.
    -1     last page.
    22     just the  23rd page.
    :-1    all but the last page.
    0:3    the first   three pages.
    -2     second-to-last page.
    :3     the first      three pages.
    -2:    last two pages.
    5:     from the sixth page onward.
    -3:-1 third & second to last.

 The third, "stride" or "step" number is also recognized.

    ::2      0 2 4 ... to the end.
    3:0:-1   3 2 1 but not 0.
    1:10:2   1 3 5 7 9
    2::-1    2 1 0.
    ::-1     all  pages in reverse order.


 Examples
     pdfly cat -o output.pdf head.pdf content.pdf :6 7: tail.pdf -1
```

```
        Concatenate all of head.pdf, all but page seven of content.pdf,
        and the last page of tail.pdf, producing output.pdf.

    pdfly cat chapter*.pdf >book.pdf
        You can specify the output file by redirection.

    pdfly cat chapter?.pdf chapter10.pdf >book.pdf
        In case you don't want chapter 10 before chapter 2.

 ─ Arguments ──────────────────────────────────────────────────────────────
 │ *    filename     PATH        [default: None] [required]                 │
 │ *    fn_pgrgs     FN_PGRGS...  filenames and/or page ranges [default: None] │
 │                                [required]                                │
 └──────────────────────────────────────────────────────────────────────────┘
 ─ Options ────────────────────────────────────────────────────────────────
 │ *  --output    -o                 PATH  [default: None] [required]       │
 │    --verbose       --no-verbose         show page ranges as they are being │
 │                                         read                             │
 │                                         [default: no-verbose]            │
 │    --help                               Show this message and exit.      │
 └──────────────────────────────────────────────────────────────────────────┘
```

3.2 Examples

3.2.1 Split a PDF

Get the second, third, and fourth page of a PDF:

```
pdfly cat input.pdf 1:4 -o out.pdf
```

3.2.2 Extract a Page

Get the sixt page of a PDF:

```
pdfly cat input.pdf 5 -o out.pdf
```

Note that it is 5, because the page indices always start at 0.

3.2.3 Concatenate two PDFs

Just combine two PDF files so that the pages come right after each other:

```
pdfly cat input1.pdf input2.pdf -o out.pdf
```

X2PDF

Convert a file to PDF.

Currently supported for "x":

- PNG
- JPG

4.1 Usage

```
$ pdfly x2pdf --help

 Usage: pdfly x2pdf [OPTIONS] X...

 Convert one or more files to PDF. Each file is a page.

─ Arguments ─────────────────────────────────────────────────────────
│ *    x       X...  [default: None] [required]                       │
│                                                                     │
─ Options ───────────────────────────────────────────────────────────
│ *  --output  -o      PATH  [default: None] [required]               │
│    --help                  Show this message and exit.              │
─────────────────────────────────────────────────────────────────────
```

4.2 Examples

4.2.1 Single file

```
$ pdfly x2pdf image.jpg -o out.pdf
$ ls -lh
-rw-rw-r-- 1 user user 47K Sep 17 21:49 image.jpg
-rw-rw-r-- 1 user user 49K Sep 17 22:48 out.pdf
```

4.2.2 Multiple files manually

```
$ pdfly x2pdf image1.jpg image2.jgp -o out.pdf
$ ls -lh
-rw-rw-r-- 1 user user 47K Sep 17 21:49 image1.jpg
-rw-rw-r-- 1 user user 15K Sep 17 21:49 image2.jpg
-rw-rw-r-- 1 user user 64K Sep 17 22:48 out.pdf
```

4.2.3 Multiple files via *

```
$ pdfly x2pdf *.jpg -o out.pdf
$ ls -lh
-rw-rw-r-- 1 user user 47K Sep 17 21:49 image1.jpg
-rw-rw-r-- 1 user user 15K Sep 17 21:49 image2.jpg
-rw-rw-r-- 1 user user 64K Sep 17 22:48 out.pdf
```

FIVE

CHANGELOG

5.1 Version 0.3.0, 2023-12-17

5.1.1 New Features (ENH)

- Add x2pdf command (#25)

5.1.2 Bug Fixes (BUG)

- boxes are floats, not int
- Add missing fpdf2 dependency (#29)

5.1.3 Documentation (DOC)

- cat command
- More examples for the cat subcommand
- Add cat subcommand
- Link to readthedocs
- Add project governance file
- Move readthedocs config file to root
- Add docs (#24)

5.1.4 Developer Experience (DEV)

- Checkout sample-files in CI (#30)
- Let dependabot update Github Actions
- Add action for automatic releases

5.1.5 Maintenance (MAINT)

- Update dependencies (#42)

- In the cat subcommand, replace the usage of the deprecated PdfMerger by PdfWriter (#34)

- Update .pre-commit-config.yaml

- Adjust x2pdf syntax

5.1.6 Testing (TST)

- cat with two files (#41)

- Test cat command with more parameters + validate result (#40)

- Adding unit tests (#28)

5.1.7 Other

- : [{'msg': 'Bump actions/setup-python from 4 to 5 (#39)', 'author': 'dependabot[bot]'}, {'msg': 'test_extract_images_monochrome() is now passing', 'author': 'CimonLucas(LCM)'}, {'msg': 'Bump actions/setup-python from 3 to 4 (#27)', 'author': 'dependabot[bot]'}, {'msg': 'Bump actions/checkout from 3 to 4 (#26)', 'author': 'dependabot[bot]'}, {'msg': 'Ensure input PDF exists for cat subcommand', 'author': 'MartinThoma'}]

Full Changelog

CONTRIBUTORS

pdfly is a free software project without any company affiliation. We cannot pay contributors, but we do value their contributions

The list might not be complete. You can find more contributors via the git history and GitHubs 'Contributors' feature.

6.1 Contributors to the pdfly project

- Thoma, Martin: Creator of pdfly. LinkedIn | StackOverflow | Blog

6.2 Adding a new contributor

Contributors are:

- Anybody who has an commit in main - no matter how big/small or how many. Also if it's via co-authored-by.
- People who opened helpful issues: (1) Bugs: with complete MCVE (2) Well-described feature requests (3) Potentially some more. The maintainers of pdfly have the last call on that one.
- Community work: This is exceptional. If the maintainers of pdfly see people being super helpful in answering issues / discussions or being very active on Stackoverflow, we also consider them being contributors to pdfly.

Contributors can add themselves or ask via an Github Issue to be added.

Please use the following format:

```
* Last name, First name: 140-characters of text; links to linkedin / github / other␣
↪profiles and personal pages are ok

OR

* GitHub Username: 140-characters of text; links to linkedin / github / other profiles␣
↪and personal pages are ok
```

and add the entry in the alphabetical order. People who . The 140 characters are everything visible after the `Name:`.

Please don't use images.

PROJECT GOVERNANCE

This document describes how the pdfly project is managed. It describes the different actors, their roles, and the responsibilities they have.

7.1 Terminology

- The **project** is pdfly - a free and open-source pure-python PDF command line tool. It includes the code, issues, and discussions on GitHub, and the documentation on ReadTheDocs, the package on PyPI.

- A **maintainer** is a person who has technical permissions to change one or more part of the projects. It is a person who is driven to keep the project running and improving.

- A **contributor** is a person who contributes to the project. That could be through writing code - in the best case through forking and creating a pull request, but that is up to the maintainer. Other contributors describe issues, help to ask questions on existing issues to make them easier to answer, participate in discussions, and help to improve the documentation. Contributors are similar to maintainers, but without technial permissions.

- A **user** is a person who imports pdfly into their code. All pdfly users are developers, but not developers who know the internals of pdfly. They only use the public interface of pdfly. They will likely have less knowledge about PDF than contributors.

- The **community** is all of that - the users, the contributors, and the maintainers.

7.2 Governance, Leadership, and Steering pdfly forward

pdfly is a free and open source project.

As pdfly does not have any formal relationship with any company and no funding, all the work done by the community are voluntary contributions. People don't get paid, but choose to spend their free time to create software of which many more are profiting. This has to be honored and respected.

pdfly has the **Benevolent Dictator** governance model. The benevolent dictator is a maintainer with all technical permissions - most importantly the permission to push new pdfly versions on PyPI.

Being benevolent, the benevolent dictator listens for decisions to the community and tries their best to make decisions from which the overall community profits - the current one and the potential future one. Being a dictator, the benevolent dictator always has the power and the right to make decisions on their own - also against some members of the community.

As pdfly is free software, parts of the community can split off (fork the code) and create a new community. This should limit the harm a bad benevolent dictator can do.

7.3 Project Language

The project language is (american) English. All documentation and issues must be written in English to ensure that the community can understand it.

We appreciate the fact that large parts of the community don't have English as their mother tongue. We try our best to understand others - automatic translators might help.

7.4 Expectations

The community can expect the following:

- The **benevolent dictator** tries their best to make decisions from which the overall community profits. The benevolent dictator is aware that his/her decisions can shape the overall community. Once the benevolent dictator notices that she/he doesn't have the time to advance pdfly, he/she looks for a new benevolent dictator. As it is expected that the benevolent dictator will step down at some point of their choice (hopefully before their death), it is NOT a benevolent dictator for life (BDFL).

- Every **maintainer** (including the benevolent dictator) is aware of their permissions and the harm they could do. They value security and ensure that the project is not harmed. They give their technical permissions back if they don't need them any longer. Any long-time contributor can become a maintainer. Maintainers can - and should! - step down from their role when they realize that they can no longer commit that time.

- Every **contributor** is aware that the time of maintainers and the benevolent dictator is limited. Short pull requests that briefly describe the solved issue and have a unit test have a higher chance to get merged soon - simply because it's easier for maintainers to see that the contribution will not harm the overall project. Their contributions are documented in the git history and in the public issues.

- Every **community member** uses a respectful language. We are all human, we get upset about things we care and other things than what's visible on the internet go on in our live. pdfly does not pay its contributors - keep all of that in mind when you interact with others. We are here because we want to help others.

7.4.1 Issues and Discussions

An issue is any technical description that aims at bringing pdfly forward:

- Bugs tickets: Something went wrong because pdfly developers made a mistake.

- Feature requests: pdfly does not support all features of the PDF specifications. There are certainly also convenience methods that would help users a lot.

- Robustness requests: There are many broken PDFs around. In some cases, we can deal with that. It's kind of a mixture between a bug ticket and a feature request.

- Performance tickets: pdfly could be faster - let us know about your specific scenario.

Any comment that is in those technial descriptions which is not helping the discussion can be deleted. This is especially true for "me too" comments on bugs or "bump" comments for desired features. People can express this with / reactions.

Discussions are open. No comments will be deleted there - except if they are clearly unrelated spam or only try to insult people (luckily, the community was very respectful so far)

7.4.2 Releases

The maintainers follow semantic versioning. Most importantly, that means that breaking changes will have a major version bump.

Be aware that unintentional breaking changes might still happen. The pdfly maintainers do their best to fix that in a timely manner - please report such issues!

7.5 People

- Martin Thoma is benevolent dictator since April 2022.

- Maintainers:

  - Matthew Stamy (mstamy2) was the benevolent dictator for a long time. He still is around on GitHub once in a while and has permissions on PyPI and GitHub.

  - Matthew Peveler (MasterOdin) is a maintainer on GitHub.

EIGHT

INDICES AND TABLES

- genindex
- modindex
- search