Appendix A. Release Notes

For a detailed list of changes, please see the file *ChangeLog* in the source distribution.

8.4.0: February 1, 2019

- Command-line Enhancements
  - *Non-compatible CLI change:* The qpdf command-line tool interprets passwords given at the command-line differently from previous releases when the passwords contain non-ASCII characters. In some cases, the behavior differs from previous releases. For a discussion of the current behavior, please see Section 3.10, “Unicode Passwords”, page 20. The incompatibilities are as follows:
    - On Windows, qpdf now receives all command-line options as Unicode strings if it can figure out the appropriate compile/link options. This is enabled at least for MSVC and mingw builds. That means that if non-ASCII strings are passed to the qpdf CLI in Windows, qpdf will now correctly receive them. In the past, they would have either been encoded as Windows code page 1252 (also known as “Windows ANSI” or as something unintelligible. In almost all cases, qpdf is able to properly interpret Unicode arguments now, whereas in the past, it would almost never interpret them properly. The result is that non-ASCII passwords given to the qpdf CLI on Windows now have a much greater chance of creating PDF files that can be opened by a variety of readers. In the past, usually files encrypted from the Windows CLI using non-ASCII passwords would not be readable by most viewers. Note that the current version of qpdf is able to decrypt files that it previously created using the previously supplied password.
    - The PDF specification requires passwords to be encoded as UTF-8 for 256-bit encryption and with PDF Doc encoding for 40-bit or 128-bit encryption. Older versions of qpdf left it up to the user to provide passwords with the correct encoding. The qpdf CLI now detects when a password is given with UTF-8 encoding and automatically transcodes it to what the PDF spec requires. While this is almost always the correct behavior, it is possible to override the behavior if there is some reason to do so. This is discussed in more depth in Section 3.10, “Unicode Passwords”, page 20.
- New options **--externalize-inline-images**, **--ii-min-bytes**, and **--keep-inline-images** control qpdf's handling of inline images and possible conversion of them to regular images. By default, **--optimize-images** now also applies to inline images. These options are discussed in Section 3.8, “Advanced Transformation Options”, page 13.
- Add options **--overlay** and **--underlay** for overlaying or underlaying pages of other files onto output pages. See Section 3.6, “Overlay and Underlay Options”, page 12 for details.
- When opening an encrypted file with a password, if the specified password doesn't work and the password contains any non-ASCII characters, qpdf will try a number of alternative passwords to try to compensate for possible character encoding errors. This behavior can be suppressed with the **--suppress-password-recovery** option. See Section 3.10, “Unicode Passwords”, page 20 for a full discussion.
- Add the **--password-mode** option to fine-tune how qpdf interprets password arguments, especially when they contain non-ASCII characters. See Section 3.10, “Unicode Passwords”, page 20 for more information.
- In the **--pages** option, it is now possible to copy the same page more than once from the same file without using the previous workaround of specifying two different paths to the same file.
- In the **--pages** option, allow use of “.” as a shortcut for the primary input file. That way, you can do **qpdf in.pdf --pages . 1-2 -- out.pdf** instead of having to repeat *in.pdf* in the command.