8.0.2: March 6, 2018

- When a loop is detected while following cross reference streams or tables, treat this as damage instead of silently ignoring the previous table. This prevents loss of otherwise recoverable data in some damaged files.
- Properly handle pages with no contents.

8.0.1: March 4, 2018

- Disregard data check errors when uncompressing **/FlateDecode** streams. This is consistent with most other PDF readers and allows qpdf to recover data from another class of malformed PDF files.
- On the command line when specifying page ranges, support preceding a page number by “r” to indicate that it should be counted from the end. For example, the range `r3-r1` would indicate the last three pages of a document.

8.0.0: February 25, 2018

- Packaging and Distribution Changes
  - QPDF is now distributed as an AppImage [https://appimage.org/] in addition to all the other ways it is distributed. The AppImage can be found in the download area with the other packages. Thanks to Kurt Pfeifle and Simon Peter for their contributions.
- Bug Fixes
  - *QPDFObjectHandle::getUTF8Val* now properly treats non-Unicode strings as encoded with PDF Doc Encoding.
  - Improvements to handling of objects in PDF files that are not of the expected type. In most cases, qpdf will be able to warn for such cases rather than fail with an exception. Previous versions of qpdf would sometimes fail with errors such as “operation for dictionary object attempted on object of wrong type”. This situation should be mostly or entirely eliminated now.
- Enhancements to the **qpdf** Command-line Tool. All new options listed here are documented in more detail in Chapter 3, *Running QPDF*, page 4.
  - The option **--linearize-pass1=*`file`*** has been added for debugging qpdf's linearization code.
  - The option **--coalesce-contents** can be used to combine content streams of a page whose contents are an array of streams into a single stream.
- API Enhancements. All new API calls are documented in their respective classes' header files. There are no non-compatible changes to the API.
  - Add function *qpdf_check_pdf* to the C API. This function does basic checking that is a subset of what **qpdf --check** performs.
  - Major enhancements to the lexical layer of qpdf. For a complete list of enhancements, please refer to the *ChangeLog* file. Most of the changes result in improvements to qpdf's ability handle erroneous files. It is also possible for programs to handle whitespace, comments, and inline images as tokens.
  - New API for working with PDF content streams at a lexical level. The new class ***QPDFObjectHandle::TokenFilter*** allows the developer to provide token handlers. Token filters can be used with several different methods in ***QPDFObjectHandle*** as well as with a lower-level interface. See comments in *QPDFObjectHandle.hh* as well as the new examples *examples/pdf-filter-tokens.cc* and *examples/pdf-count-strings.cc* for details.