

- When encrypting with 128-bit and 256-bit encryption, new encryption options **--assemble**, **--annotate**, **--form**, and **--modify-other** allow more fine-grained granularity in configuring options. Before, the **--modify** option only configured certain predefined groups of permissions.
- Bug Fixes and Enhancements
 - *Potential data-loss bug*: Versions of qpdf between 8.1.0 and 8.3.0 had a bug that could cause page splitting and merging operations to drop some font or image resources if the PDF file's internal structure shared these resource lists across pages and if some but not all of the pages in the output did not reference all the fonts and images. Using the **--preserve-unreferenced-resources** option would work around the incorrect behavior. This bug was the result of a typo in the code and a deficiency in the test suite. The case that triggered the error was known, just not handled properly. This case is now exercised in qpdf's test suite and properly handled.
 - When optimizing images, detect and refuse to optimize images that can't be converted to JPEG because of bit depth or color space.
 - Linearization and page manipulation APIs now detect and recover from files that have duplicate Page objects in the page tree.
 - Using older option **--stream-data=compress** with object streams, object streams and xref streams were not compressed.
 - When the tokenizer returns inline image tokens, delimiters following ID and EI operators are no longer excluded. This makes it possible to reliably extract the actual image data.
- Library Enhancements
 - Add method *QPDFPageObjectHelper::externalizeInlineImages* to convert inline images to regular images.
 - Add method *QUtil::possible_repaired_encodings()* to generate a list of strings that represent other ways the given string could have been encoded. This is the method the QPDF CLI uses to generate the strings it tries when recovering incorrectly encoded Unicode passwords.
 - Add new versions of *QPDFWriter::setR{3,4,5,6}EncryptionParameters* that allow more granular setting of permissions bits. See *QPDFWriter.hh* for details.
 - Add new versions of the transcoders from UTF-8 to single-byte coding systems in **QUtil** that report success or failure rather than just substituting a specified unknown character.
 - Add method *QUtil::analyze_encoding()* to determine whether a string has high-bit characters and is appears to be UTF-16 or valid UTF-8 encoding.
 - Add new method *QPDFPageObjectHelper::shallowCopyPage()* to copy a new page that is a “shallow copy” of a page. The resulting object is an indirect object ready to be passed to *QPDFPageDocumentHelper::addPage()* for either the original **QPDF** object or a different one. This is what the **qpdf** command-line tool uses to copy the same page multiple times from the same file during splitting and merging operations.
 - Add method *QPDF::getUniqueId()*, which returns a unique identifier for the given QPDF object. The identifier will be unique across the life of the application. The returned value can be safely used as a map key.
 - Add method *QPDF::setImmediateCopyFrom*. This further enhances qpdf's ability to allow a **QPDF** object from which objects are being copied to go out of scope before the destination object is written. If you call this method on a **QPDF** instances, objects copied *from* this instance will be copied immediately instead of lazily. This option uses more memory but allows the source object to go out of scope before the destination object is written in all cases. See comments in *QPDF.hh* for details.