

- Image comparison tests are off by default now since they are not needed to verify a correct build or port of qpdf. They are needed only when changing the actual PDF output generated by qpdf. You should enable them if you are making deep changes to qpdf itself. See *README.md* for details.
- Large file tests are off by default but can be turned on with `./configure` or by setting an environment variable before running the test suite. See *README.md* for details.
- When qpdf's test suite fails, failures are not printed to the terminal anymore by default. Instead, find them in *build/qtest.log*. For packagers who are building with an autobuilder, you can add the `--enable-show-failed-test-output` option to `./configure` to restore the old behavior.

2.3.1: December 28, 2011

- Fix thread-safety problem resulting from non-thread-safe use of the PCRE library.
- Made a few minor documentation fixes.
- Add workaround for a bug that appears in some versions of ghostscript to the test suite
- Fix minor build issue for Visual C++ 2010.

2.3.0: August 11, 2011

- Bug fix: when preserving existing encryption on encrypted files with cleartext metadata, older qpdf versions would generate password-protected files with no valid password. This operation now works. This bug only affected files created by copying existing encryption parameters; explicit encryption with specification of cleartext metadata worked before and continues to work.
- Enhance **QPDFWriter** with a new constructor that allows you to delay the specification of the output file. When using this constructor, you may now call `QPDFWriter::setOutputFilename` to specify the output file, or you may use `QPDFWriter::setOutputMemory` to cause **QPDFWriter** to write the resulting PDF file to a memory buffer. You may then use `QPDFWriter::getBuffer` to retrieve the memory buffer.
- Add new API call `QPDF::replaceObject` for replacing objects by object ID
- Add new API call `QPDF::swapObjects` for swapping two objects by object ID
- Add `QPDFObjectHandle::getDictAsMap` and `QPDFObjectHandle::getArrayAsVector` to allow retrieval of dictionary objects as maps and array objects as vectors.
- Add functions `qpdf_get_info_key` and `qpdf_set_info_key` to the C API for manipulating string fields of the document's `/Info` dictionary.
- Add functions `qpdf_init_write_memory`, `qpdf_get_buffer_length`, and `qpdf_get_buffer` to the C API for writing PDF files to a memory buffer instead of a file.

2.2.4: June 25, 2011

- Fix installation and compilation issues; no functionality changes.

2.2.3: April 30, 2011

- Handle some damaged streams with incorrect characters following the stream keyword.
- Improve handling of inline images when normalizing content streams.
- Enhance error recovery to properly handle files that use object 0 as a regular object, which is specifically disallowed by the spec.