integer type coercion occurs have been scrutinized. A new casting policy has been documented in the manual. This is of concern mainly to people porting qpdf to new platforms or compilers. It is not visible to programmers writing code that uses the library

  - Some internal limits have been removed in code that converts numbers to strings. This is largely invisible to users, but it does trigger a bug in some older versions of mingw-w64's C++ library. See *README-windows.md* in the source distribution if you think this may affect you. The copy of the DLL distributed with qpdf's binary distribution is not affected by this problem.

- The RPM spec file previously included with qpdf has been removed. This is because virtually all Linux distributions include qpdf now that it is a dependency of CUPS filters.

- A few bug fixes are included:

  - Overridden compressed objects are properly handled. Before, there were certain constructs that could cause qpdf to see old versions of some objects. The most usual manifestation of this was loss of filled in form values for certain files.

  - Installation no longer uses GNU/Linux-specific versions of some commands, so **make install** works on Solaris with native tools.

  - The 64-bit mingw Windows binary package no longer includes a 32-bit DLL.

4.0.1: January 17, 2013

- Fix detection of binary attachments in test suite to avoid false test failures on some platforms.

- Add clarifying comment in *QPDF.hh* to methods that return the user password explaining that it is no longer possible with newer encryption formats to recover the user password knowing the owner password. In earlier encryption formats, the user password was encrypted in the file using the owner password. In newer encryption formats, a separate encryption key is used on the file, and that key is independently encrypted using both the user password and the owner password.

4.0.0: December 31, 2012

- Major enhancement: support has been added for newer encryption schemes supported by version X of Adobe Acrobat. This includes use of 127-character passwords, 256-bit encryption keys, and the encryption scheme specified in ISO 32000-2, the PDF 2.0 specification. This scheme can be chosen from the command line by specifying use of 256-bit keys. qpdf also supports the deprecated encryption method used by Acrobat IX. This encryption style has known security weaknesses and should not be used in practice. However, such files exist "in the wild," so support for this scheme is still useful. New methods *QPDFWriter::setR6EncryptionParameters* (for the PDF 2.0 scheme) and *QPDFWriter::setR5EncryptionParameters* (for the deprecated scheme) have been added to enable these new encryption schemes. Corresponding functions have been added to the C API as well.

- Full support for Adobe extension levels in PDF version information. Starting with PDF version 1.7, corresponding to ISO 32000, Adobe adds new functionality by increasing the extension level rather than increasing the version. This support includes addition of the *QPDF::getExtensionLevel* method for retrieving the document's extension level, addition of versions of *QPDFWriter::setMinimumPDFVersion* and *QPDFWriter::forcePDFVersion* that accept an extension level, and extended syntax for specifying forced and minimum versions on the command line as described in Section 3.8, "Advanced Transformation Options", page 13. Corresponding functions have been added to the C API as well.

- Minor fixes to prevent qpdf from referencing objects in the file that are not referenced in the file's overall structure. Most files don't have any such objects, but some files have contain unreferenced objects with errors, so these fixes prevent qpdf from needlessly rejecting or complaining about such objects.