  - Add new methods *QPDFAcroFormDocumentHelper::generateAppearancesIfNeeded* and *QPDFFormFieldObjectHelper::generateAppearance* for generating appearance streams. See discussion in *QPDFFormFieldObjectHelper.hh* for limitations.

  - Add two new helper functions for dealing with resource dictionaries: *QPDFObjectHandle::getResourceNames()* returns a list of all second-level keys, which correspond to the names of resources, and *QPDFObjectHandle::mergeResources()* merges two resources dictionaries as long as they have non-conflicting keys. These methods are useful for certain types of objects that resolve resources from multiple places, such as form fields.

  - Add methods *QPDFPageDocumentHelper::flattenAnnotations()* and *QPDFAnnotationObjectHelper::getPageContentForAppearance()* for handling low-level details of annotation flattening.

  - Add new helper classes: ***QPDFOutlineDocumentHelper***, ***QPDFOutlineObjectHelper***, ***QPDFPageLabelDocumentHelper***, ***QPDFNameTreeObjectHelper***, and ***QPDFNumberTreeObjectHelper***.

  - Add method *QPDFObjectHandle::getJSON()* that returns a JSON representation of the object. Call *serialize()* on the result to convert it to a string.

  - Add a simple JSON serializer. This is not a complete or general-purpose JSON library. It allows assembly and serialization of JSON structures with some restrictions, which are described in the header file. This is the serializer used by qpdf's new JSON representation.

  - Add new ***QPDFObjectHandle::Matrix*** class along with a few convenience methods for dealing with six-element numerical arrays as matrices.

  - Add new method *QPDFObjectHandle::wrapInArray*, which returns the object itself if it is an array, or an array containing the object otherwise. This is a common construct in PDF. This method prevents you from having to explicitly test whether something is a single element or an array.

- Build Improvements

  - It is no longer necessary to run **autogen.sh** to build from a pristine checkout. Automatically generated files are now committed so that it is possible to build on platforms without autoconf directly from a clean checkout of the repository. The **configure** script detects if the files are out of date when it also determines that the tools are present to regenerate them.

  - Pull requests and the master branch are now built automatically in Azure Pipelines [https://dev.azure.com/qpdf/qpdf/_build], which is free for open source projects. The build includes Linux, mac, Windows 32-bit and 64-bit with mingw and MSVC, and an AppImage build. Official qpdf releases are now built with Azure Pipelines.

- Notes for Packagers

  - A new section has been added to the documentation with notes for packagers. Please see Section 2.3, “Notes for Packagers”, page 3.

  - The qpdf detects out-of-date automatically generated files. If your packaging system automatically refreshes libtool or autoconf files, it could cause this check to fail. To avoid this problem, pass **--disable-check-autofiles** to **configure**.

  - If you would like to have qpdf completion enabled automatically, you can install completion files in the distribution's default location. You can find sample completion files to install in the *completions* directory.