
Chapter 5. Using the QPDF Library

5.1. Using QPDF from C++

The source tree for the qpdf package has an *examples* directory that contains a few example programs. The *qpdf/qpdf.cc* source file also serves as a useful example since it exercises almost all of the qpdf library's public interface. The best source of documentation on the library itself is reading comments in *include/qpdf/QPDF.hh*, *include/qpdf/QPDFWriter.hh*, and *include/qpdf/QPDFObjectHandle.hh*.

All header files are installed in the *include/qpdf* directory. It is recommend that you use `#include <qpdf/QPDF.hh>` rather than adding *include/qpdf* to your include path.

When linking against the qpdf static library, you may also need to specify `-lz -ljpeg` on your link command. If your system understands how to read *libtool .la* files, this may not be necessary.

The qpdf library is safe to use in a multithreaded program, but no individual QPDF object instance (including QPDF, QPDFObjectHandle, or QPDFWriter) can be used in more than one thread at a time. Multiple threads may simultaneously work with different instances of these and all other QPDF objects.

5.2. Using QPDF from other languages

The qpdf library is implemented in C++, which makes it hard to use directly in other languages. There are a few things that can help.

“C”

The qpdf library includes a “C” language interface that provides a subset of the overall capabilities. The header file *qpdf/qpdf-c.h* includes information about its use. As long as you use a C++ linker, you can link C programs with qpdf and use the C API. For languages that can directly load methods from a shared library, the C API can also be useful. People have reported success using the C API from other languages on Windows by directly calling functions in the DLL.

Python

A Python module called [pikepdf](https://pypi.org/project/pikepdf/) [https://pypi.org/project/pikepdf/] provides a clean and highly functional set of Python bindings to the qpdf library. Using pikepdf, you can work with PDF files in a natural way and combine qpdf's capabilities with other functionality provided by Python's rich standard library and available modules.

Other Languages

Starting with version 8.3.0, the **qpdf** command-line tool can produce a JSON representation of the PDF file's non-content data. This can facilitate interacting programmatically with PDF files through qpdf's command line interface. For more information, please see [Chapter 6, QPDF JSON](#), page 25.