

Foreword

As PyQt's creator, I'm delighted to see that this book has been written. Although I served as one of the book's technical reviewers, I'm happy to confess that I learned a few things myself.

The PyQt documentation covers the APIs of all the PyQt classes. This book shows you how to use all those classes, how to combine them to create dialogs, main windows, and entire applications—all of which look good and work well, with no arbitrary limits, and using a programming language that is a joy to use.

What I particularly like about the book is that the examples aren't trivial ones designed to illustrate a simple point, but are potentially useful in their own right. The way that different approaches are considered will reward the reader who wants to develop a deeper understanding of how to apply PyQt to the development of large scale, production quality applications.

I began the PyQt story back in the late 1990s. I had been using Tcl/Tk for some time, but I felt that Tk applications looked ugly, especially when I saw what had been achieved with the first version of KDE. I had wanted to switch to Python, and so I thought I would combine the change of language with a change of GUI library.

Initially I used some wrappers that had been written using SWIG, but I concluded that I could produce a more suitable wrapper tool myself. I set to work creating SIP, and released PyQt 0.1 supporting Qt 1.41 in November 1998. Development has continued regularly ever since, both to keep up with new releases of Qt and to broaden the scope of PyQt with, for example, the addition of support tools and improved documentation. By 2000, PyQt 2.0 supported Qt 2.2 on both Linux and Windows. Qt 3 support appeared in 2001, and Mac OS X support in 2002. The PyQt4 series began with PyQt 4.0 in June 2006 with support for Qt 4.

My primary goal has always been to allow Python and Qt to work together in a way that feels natural to Python programmers, while allowing them to do anything they want in Python that can be done in C++. The key to achieving this was the development of SIP. This gave me a specialized code generator over which I had complete control and ensures that Python and Qt will always fit snugly together.

The essential process of developing and maintaining PyQt is now well established. Much of the work is now automated, which means that keeping up with

new releases of Qt from Trolltech is no longer the problem it once was, and ensures that PyQt will continue for years to come.

It's been very gratifying to watch the growth of the PyQt community over the years. If this book is part of your introduction to PyQt, then welcome!

— Phil Thompson
Wimborne, Dorset, U.K.
August 25, 2007