

Rapid GUI Application Development with Python

Volker Kuhlmann

Kiwi PyCon 2009 — Christchurch

7 November 2009

Copyright © 2009 by Volker Kuhlmann

Released under Creative Commons Attribution Non-commercial Share-alike 3.0

<http://volker.top.geek.nz/linux/presentation/RapidGUIwithPython.pdf>

Introduction

- Make a GUI desktop application
 - Application often is means to an end, not the end itself
 - The quicker the better...
- What is needed to do this?

Overview

- Python interpreter
- GUI toolkit
- IDE
- GUI builder
- Debugger
- Database
- Testing
- Application packaging
- Looking over the fence
- Summary
- Future

Python interpreter

- Install stable release for your platform
 - Already packaged up
- Current stable versions 2.6, 3.1
- 3.x is not totally backwards compatible

GUI toolkit

- Provides graphical user interface elements (widgets)
 - Buttons, scrollbars, text boxes
 - Tree view lists
 - File-open dialogs
 - ...
- Many of these available as system library
 - Best known: GTK, Qt
- Some platforms only provide a single built-in one
- Python wrapper needed for use with Python
- Understanding of the toolkit itself needed to some extent

Python GUI toolkit choices

- Tkinter — Built into Python on all platforms
 - Free. Well-aged, limited, somewhat basic. Universally available.
- wxPython
 - Free, stable, cross-platform, large number of widgets
- PyGTK
 - Free, stable, cross-platform
- PyQt
 - Free only for GPL'ed applications (Qt LGPL since 4.5)
 - Stable, cross-platform, very advanced; HTML-rendering in text boxes
- PySide
 - Free replacement for PyQt, new, not yet mature, sponsored by Nokia.
- Others — not first choice in general

IDE

- Kind of a workbench
- Editor – of course
 - Code completion
 - Calltips
- Integrating:
 - Documentation
 - Debugger
 - Hover over variable to inspect value
 - Code profiler
 - Report generators
 - Code + DB structures
 - Documentation
 - Revision control system interface

IDE choices

- Eclipse / PyDev
 - FOSS, well supported
- BoaConstructor
 - With GUI builder
- Eric
 - With GUI builder (Qt Designer)
- Wing IDE — commercial (free for established FOSS projects)
- Komodo — commercial
- Spyder
 - Aimed at science users
- Emacs/vim

GUI Builder

- Use a GUI to design a GUI
- wxPython
 - BoaConstructor
 - XRCed
 - XML editor, fast results with good knowledge of wxPython
 - WxDesigner (commercial)
- PyGTK
 - wxGlade / Autoglade
 - Gazpacho
- PyQt / PySIDE
 - Qt Designer
 - Mature, well-designed
 - Integrated with Eric

Debugger

- Integrated into IDEs:
 - Eclipse/PyDev
 - FOSS
 - Komodo, Wing IDE
- Stand-alone:
 - winpdb
 - GUI to command line debuggers, incl. pydb
 - <http://winpdb.org/about>
 - ddd
 - GUI to command line debuggers, incl. pydb
 - Very good
 - GPL, Unix/Linux only

Database

- Bindings exist to interface to a number of databases
- SQLite
 - Small, embedded or single program, lightweight
- MySQL, PostgreSQL
 - DBs need to be set up separately, heavy duty
- Uniform Python SQL database API
- Higher-level abstraction layer desirable
 - No need to write and debug SQL code
- buzbug — pure Python DB implementation
 - Slow, but Python-like interface, no SQL

Testing

- Large number of stand-alone solutions for different purposes and methodologies
- Unit testing, black box/white box testing
- Extensive overview:
<http://pycheesecake.org/wiki/PythonTestingToolsTaxonomy>

Application packaging

- **setuptools**
 - Use Python interpreter to package up, and install, Python modules.
 - Not under system package manager control.
 - Needs privileges to install for all users.
 - Platform-independent.
- **Package for the system package handler (e.g. rpm, deb)**
 - Tedious
 - Platform-specific
 - Only on some platforms
- **Executable installer**
 - Security risks?
 - Platform-specific
 - Only on some platforms

Looking over the fence

- Always good to see what others are up to
- Fully integrated IDEs with all mentioned features are available for \$x00.–
 - Not for Python though... 😊

Summary

- Choose GUI toolkit and IDE carefully
 - Toolkit suitability for target application
 - Toolkit license
 - IDE feature set
 - Consider commercial options
- Good-looking native applications can be made with a scripting language
- Installing the full tool chain is tedious
 - Finding out and deciding what is needed takes too long
- Consider commercial options for part(s) of the tool chain

Future

- Desirable:
 - Better integration of tool chain
 - Tool improvements

References / Reading

- Polo, Guilherme: PyGTK, PyQt, Tkinter and wxPython comparison. The Python Papers, Volume 3 Issue 1, pp 26ff.
- Hetland, Magnus L.: Beginning Python. From Novice to Professional. 2nd ed. Apress.
 - Good for learning Python programming.
- Summerfield, Mark: Rapid GUI Programming with Python and QT. Prentice Hall.
 - Guide, by a Qt developer. (Could be clearer written.)
- <http://python.org/>
 - Lists available software by categories
- Wikipedia (for IDEs, GUI toolkits, etc)
- Your favourite search engine