

Programming Qt + KDE with Python

Simon Edwards <simonedwards@kde.nl>

What is Python?

- High level, interpreted programming language.
- Features:
 - Simple & concise syntax
 - Built in lists, hash/dictionary datatypes
 - Strong dynamic typing
 - Object oriented
 - Garbage collection
 - Exceptions and stacktraces
 - Meta-programming + generators
 - Large standard library
- Benevolent dictator: Guido van Rossum.

What I've been doing with it...

Disk & Filesystems - Disk & Filesystem Configuration

Available Disks and Filesystems:

Name	Mount Point	Type	Device	Enabled
Disk Hitachi HTS54161				
1 Partition 4.9 Gb			/dev/sda1	
2 Partition 53.2 Gb	/media/windows	vfat	/dev/sda2	● Disabled
3 Partition 2.0 Gb	none	swap	/dev/sda3	● Enabled
4 Partition 51.7 Gb	/	jfs	/dev/sda4	● Enabled
System				
Burner DVD/CDRW GSA-T10N	/media/data1	cifs	\\192.168.1.1\data	● Disabled

Buttons: New..., Modify..., Delete..., Enable, Disable, Details..., About, Close

Display Configuration - Display and Graphics Configuration


Size, Orientation & Positioning | Color & Gamma | Hardware | Power saving

Monitor Orientation

- Normal
- Left edge on top
- Right edge on top
- Upsidedown

Second screen

Screen size

Lower  Higher

1280 x 800

Refresh: 60 Hz

Buttons: About, Reset, Apply, Close

What are language bindings?

- Glue between the object model, type system, calling conventions etc from one language to another. (KDE) C++ «» Python in this case.
- Components:
 - SIP: binding generator.
 - PyQt: bindings for Qt.
 - PyKDE: bindings for KDE.
- GPL, (commercial licenses available)
- Thanks go to Phil Thompson and Jim Bublitz.

KDE 3 Example

```
#!/usr/bin/python
# GPL. (C) 2005 Simon Edwards.
from qt import *
from kdecore import *
from kdeui import *
import sys

aboutdata = KAboutData("test","test","1.0", "A test", \
    KAboutData.License_GPL, "Copyright (C) 2005-7 Simon Edwards")
KCmdLineArgs.init(sys.argv,aboutdata)
kapp = KApplication()

label = QLabel("aKademy 2007",None)
label.adjustSize()
label.show()

kapp.setMainWidget(label)
kapp.exec_loop()
```

DCOP Example

- KDE's interprocess communication system.

```
#!/usr/bin/python
# GPL. (C) 2005 Simon Edwards.
import sys
import dcopext
from kdecore import *

aboutdata = KAboutData("dcop_test","dcop_test","1.0", "A test",
    KAboutData.License_GPL, "Copyright (C) 2005-7 Simon Edwards")
KCmdLineArgs.init(sys.argv,aboutdata)
app = Kapplication()

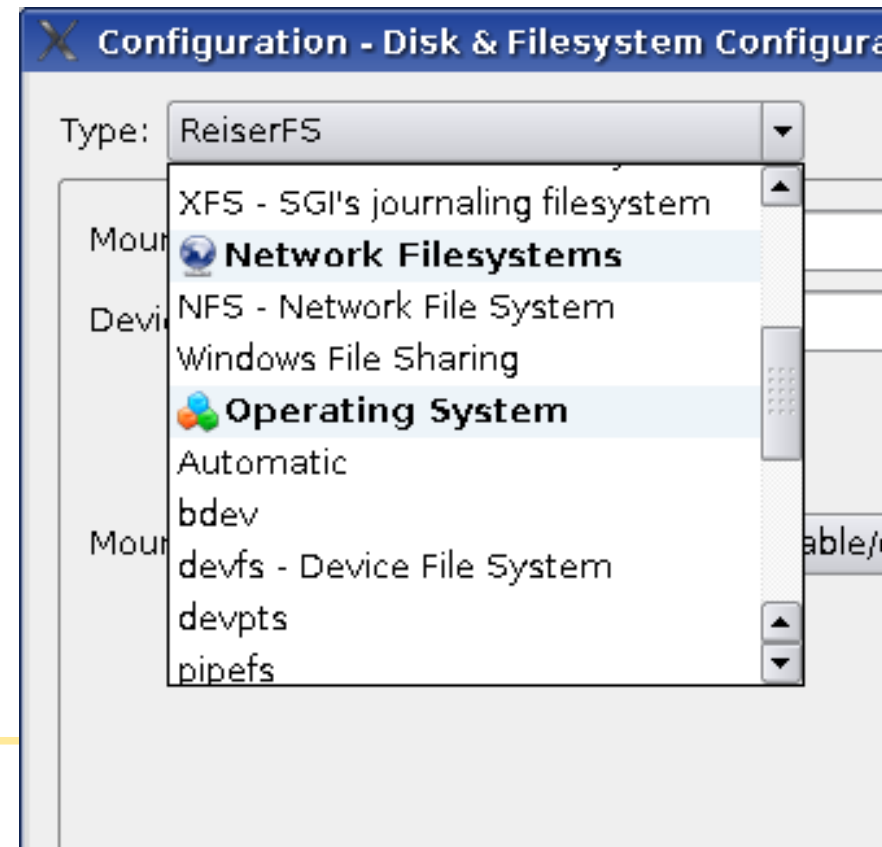
dcop = app.dcopClient()
kwin = dcopext.DCOPApp("kicker", dcop)
ok,size = kwin.Panel.panelSize()
print "Panel size is:",size
```

Subclass Example

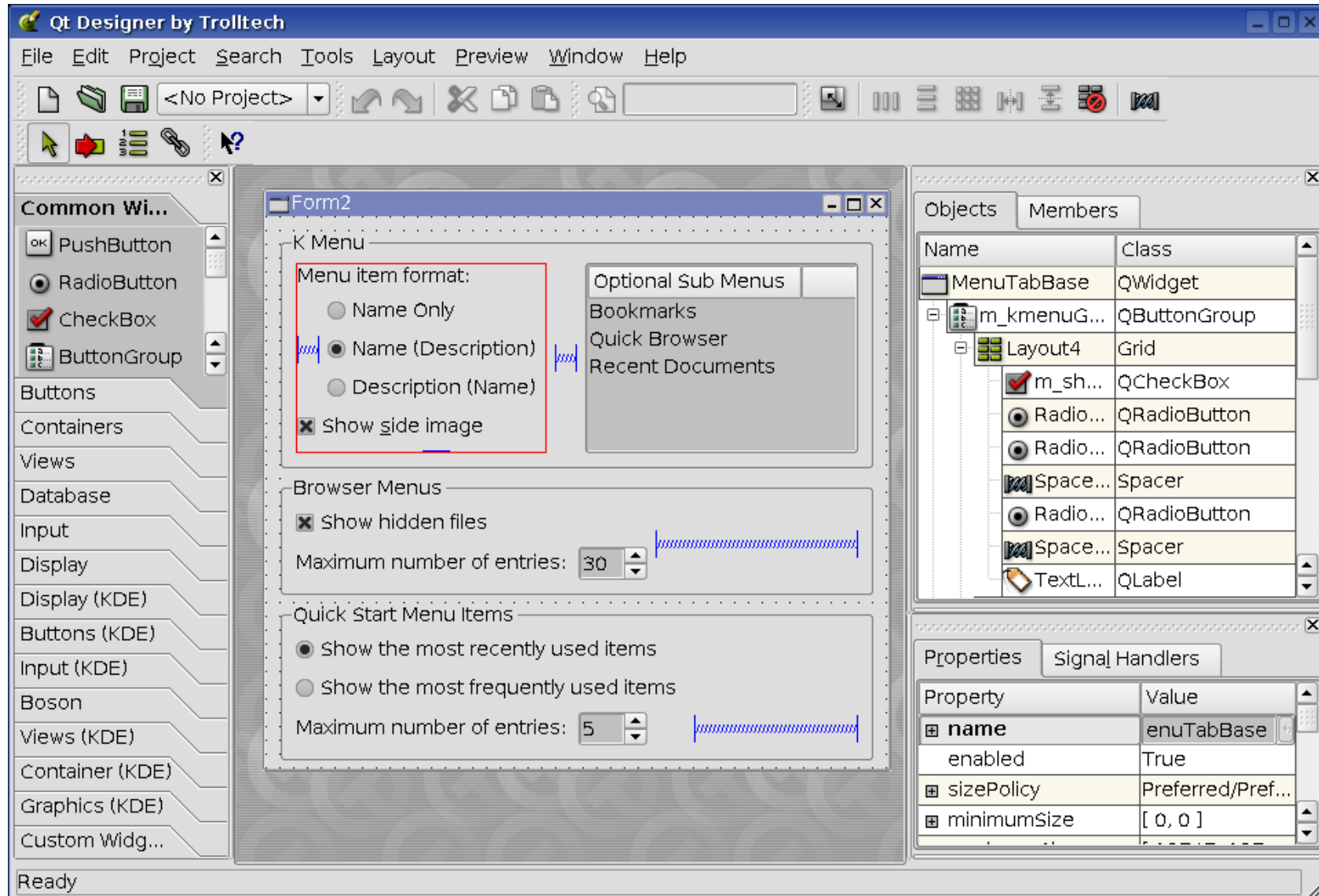
```
# A read-only ListBox item that also uses the 'alternate' background  
# colour as background.
```

```
class ROListBoxItem(QListBoxPixmap):  
    def __init__(self, listbox, pix, text):  
        QListBoxPixmap.__init__(self, listbox, pix, text)  
        self.bgc = KglobalSettings.alternateBackgroundColor()  
        self.setSelectable(False)
```

```
def paint(self, p):  
    boldfont = QFont(p.font())  
    boldfont.setWeight(QFont.Bold)  
    p.setFont(boldfont)  
    p.setBackgroundColor(self.bgc)  
    p.eraseRect(0, 0, \  
                self.listBox().width(), \  
                self.height(self.listBox()))  
    QListBoxPixmap.paint(self, p)
```



Qt-Designer

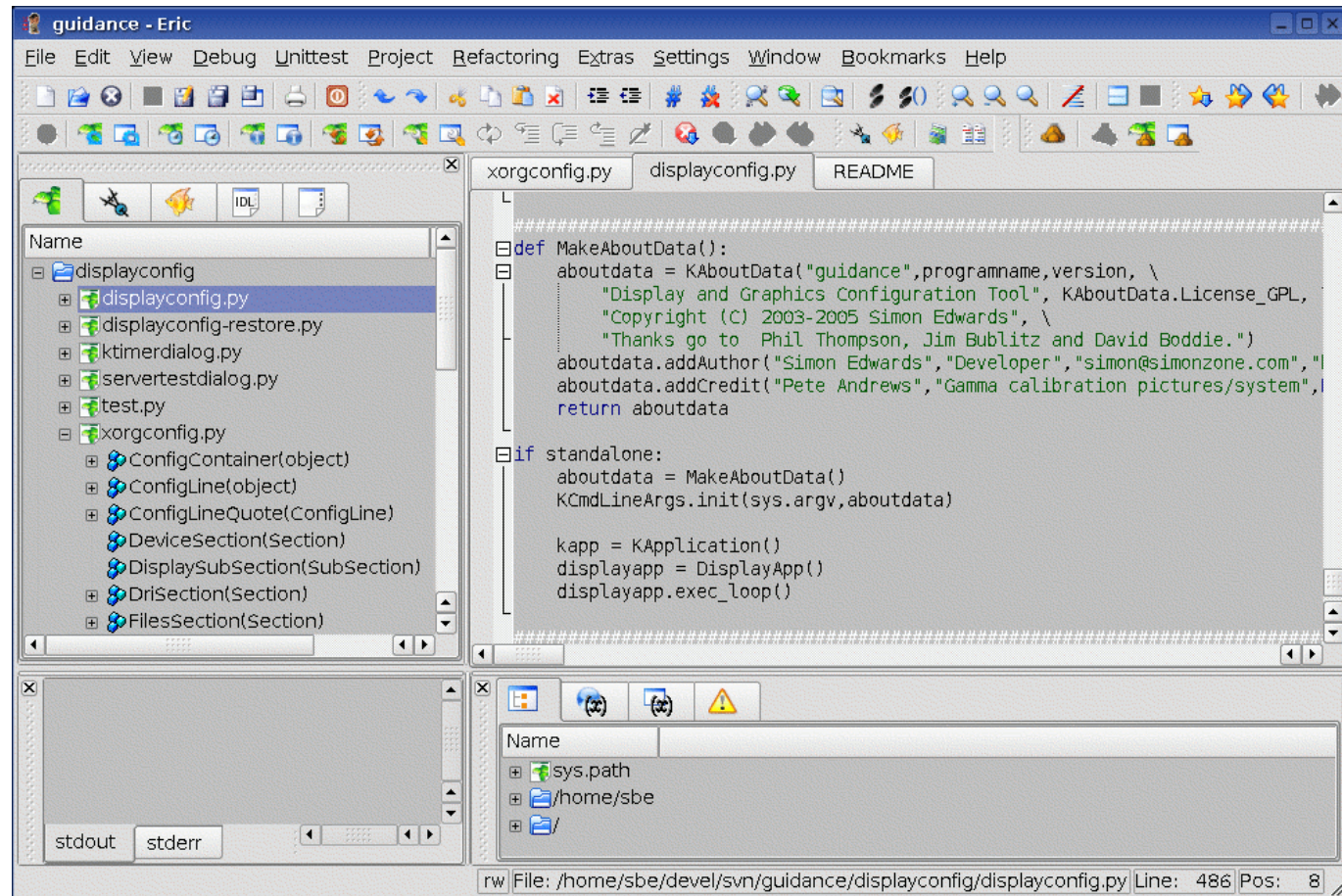


Performance & Cost

- GUI programs are typically I/O bound by the user, network, harddisk etc.
- Qt/KDE (C++) handles the GUI, menus, widgets etc.
- Overhead: Python Interpreter, PyQt and PyKDE.

Development Tools

- Eric3: Python IDE, using PyQt.
- Debugger.
- Unit testing.
- Version Control.
- Refactoring.
- Coverage.



Embedding & Plugins

Disk & Filesystems - Configure your desktop

File View Settings Help

Index Search Help

Disk & Filesystems

Available Disks and Filesystems:

Name	Mount Point	Type	Device	Enabled
Optical Disk 'LITE-ON LTR-16102B'				
System				
Disk 'Maxtor 6Y120L0'				
1 Partition 20.0 Gb	/mnt/win_c	ntfs	/dev/hda1	Enabled
2 Partition 20.0 Gb	/	reiserfs	/dev/hda2	Enabled
3 Partition 1.9 Gb	swap	swap	/dev/hda3	Enabled
4 Partition 1 Kb			/dev/hda4	
5 Partition 27.9 Gb	/mnt/data	vfat	/dev/hda5	Enabled
6 Partition 44.6 Gb	/mnt/data2	reiserfs	/dev/hda6	Enabled
/mnt/cdrom	/mnt/cdrom	auto	/dev/hdc	Disabled
/mnt/floppy	/mnt/floppy	supermount	none	Enabled
/mnt/blik_d	/mnt/blik_d	smbfs	\\blik\D\$	Disabled

New... Modify... Delete... Enable Disable

Side effects

- Rapid development
- Rapid debugging. Useful bug reports with exceptions, stacktraces and line numbers.
- Lowers the *tinkering* barrier
- ...and hence lowers the barrier for contributors
- Puts the fun back in development

Current Status

- PyQt 4.0 for Qt 4.1 released 10 June 2006
- Current version is PyQt v4.2 (Qt 4.2)
- Snapshots support Qt 4.3.
- PyKDE beta release expected around the first KDE 4 beta release.

Book

“Rapid GUI Programming with Python and Qt” by
Mark Summerfield

- Covers Python 2.5, PyQt 4.2 & Qt 4.2.
- On safari.informit.com
- Printed version coming end October in the US.
- ~550 page.
- (Disclaimer: I haven't read it, but it sounds promising.)

Other Python Books

- “A Byte of Python” - beginners Python book
- “Dive into Python” - intermediate Python book