wxPython in a Nutshell

Robin Dunn

http://wxPython.org/

O’Reilly Open Source Convention
July 26–30, 2004
The best way to eat an elephant...
...is one bite at a time
Introduction to wxPython

• wxPython is a GUI toolkit for Python, built upon the wxWidgets C++ toolkit.
  – Cross platform: Windows, Linux, Unix, OS X.
  – Uses native widgets/controls, plus many platform independent widgets.

• Mature, well established projects.
  – wxWidgets: 1992
  – wxPython: 1996
Introduction: architecture

Proxy classes → wxPython Extension Modules → wxWidgets Toolkit → Platform GUI → Operating System

wxPython: Cross Platform GUI Toolkit
Introduction: partial class hierarchy

wx.Object \rightarrow wx.EvtHandler \rightarrow wx.Window

wx.TopLevelWindow \rightarrow wx.Frame

wx.Panel \rightarrow wx.Dialog

wx.ScrolledWindow

wx.Control
Getting started with wxPython

• Installation is simple -- binary installers are available at SourceForge and via http://wxPython.org/download.php for:
  – Windows: *.exe
  – Linux:   *.rpm (and *.deb’s are available separately.)
  – OS X:    *.dmg, a disk image that contains an Installer package.
• Can be built from source for other Unix-like systems.
Getting started with wxPython

• Choose an installer.
• Which version of Python do you use?
  – 2.2, or 2.3
• Unicode?
  – Windows, but be careful with Win9x/ME
  – Linux/Unix, with the GTK2 build
  – OS X, soon
• or ANSI?
  – All platforms
Getting started with wxPython

• Choose an editor or development environment:
  – Boa Constructor
  – WingIDE
  – PyAlaMode
  – SCiTE
  – Emacs, vi, etc.

• It’s just plain text, so an ordinary editor and command line will do.
Getting started with wxPython

• Ready, set, go!
• The wxPython Demo is a great way to learn about the capabilities of the toolkit.
Getting started with wxPython

Python

Python is an interpreted, interactive, object-oriented programming language often compared to Tcl, Perl, Scheme, or Java.

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, and new built-in modules are easily written in C or C++.

Python is also usable as an extension language for applications that need a programmable interface.

wxWindows

wxWindows is a free C++ framework designed to make cross-platform programming child's play. Well, almost. wxWindows 2 supports Windows 3.1, 95, 98/NT, Unix with GTK/Motif/Leastif, with a Mac version underway. Other ports are under consideration.

wxWindows is a set of libraries that allows C++ applications to compile and run on several different types of computers, with minimal source code changes. There is one library per supported GUI (such as Motif, or Windows). As well as providing a common API (Application Programming Interface) for GUI functionality, it provides functionality for accessing some commonly-used operating system facilities, such as copying or deleting files. wxWindows is a framework in the sense that it provides a lot of built-in functionality, which the application can

wxPython: Cross Platform GUI Toolkit
Getting started with wxPython

Python

Python is an interpreted, interactive, object-oriented programming language often compared to Tcl, Perl, Scheme, or Java.

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, and new built-in modules are easily written in C or C++. Python is also usable as an extension language for applications that need a programmable interface.

wxWindows

wxWindows is a free C++ framework designed to make cross-platform programming child's play. Well, almost. wxWindows 2 supports Windows 3.1/95/98/NT, Unix with GTK/Motif/Qt, and even a Mac version underway. Other ports are under consideration.

wxWindows is a set of libraries that allows C++ applications to compile and run on several different types of computers, with minimal source code changes. There is one library per supported GUI (such as Motif or Windows). As well as providing a common API (Application Programming Interface) for GUI functionality, it provides functionality for accessing some commonly used operating system facilities, such as copying or deleting files. wxWindows is a 'framework' in the sense that it provides a lot of built-in

window handle: 60617469
Getting started with wxPython

Python

Python is an interpreted, interactive, object-oriented programming language often compared to Tcl, Perl, Scheme, or Java.

Python combines remarkable power with very clear syntax. It has modules, classes, exceptions, very high level dynamic data types, and dynamic typing. There are interfaces to many system calls and libraries, and new built-in modules are easily written in C or C++. Python is also usable as an extension language for applications that need a programmable interface.

wxWindows

wxWindows is a free C++ framework designed to make cross-platform programming child’s play. Well, almost. wxWindows 2 supports Windows 3.1/95/98/NT, Unix with GTK/Motif/Lasstif, with a Mac version underway. Other ports are under consideration.

wxWindows is a set of libraries that allows C++ applications to compile and run on several different types of computers, with minimal source code changes.
Widgets

- `wx.Button`, `wx.BitmapButton`
- `wx.RadioBox`, `wx.RadioButton`
- `wx.CheckBox`
- `wx.Choice`
- `wx.ComboBox`
- `wx.SpinButton`
Widgets

- `wx.ToggleButton`
- `wx.gizmosEditableListBox`
- `wx.lib.masked.TextCtrl`
- `wx.calendar.CalendarCtrl`
- `wx.lib.masked.TimeCtrl`
Widgets

• `wx.TextCtrl`
  - Password masking, multi-line with or without word-wrap, simple attributes, etc.
Widgets

- `wx.ListBox`
- `wx.CheckListBox`
- `wx.Gauge`
- `wx.Slider`
- `wx.StaticBox`
Widgets

wxPython: Cross Platform GUI Toolkit
Widgets

- `wx.StatusBar`

- `wx.ToolBar`
Widgets

- **wx.Notebook**
  - Manages multiple windows with tabs.
  - Tabs can be on any side of the notebook that the platform supports.
Widgets

- **wx.ListCtrl**
  - Supports list, icon, small icon, report views.
  - Virtual mode, where data items are provided by overloaded methods.
Widgets

• `wx.TreeCtrl`
  - Supports images for various node states.
  - Can be virtualized by delaying the adding of child items until the parent is expanded.
Widgets

- **wx.SplitterWindow**
  - Can be split vertically or horizontally.
  - Draggable sash for redistributing the space between sub-windows.

![Splitter Window Diagram](image)
Widgets

![Simple Grid Demo](image)

<table>
<thead>
<tr>
<th></th>
<th>Custom</th>
<th>column</th>
<th>labels</th>
<th>D</th>
<th>E</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>First cell</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Limited text</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>Another cell</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>Yet another cell</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td>This cell is read-only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>123</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>123.34</td>
<td></td>
<td></td>
<td></td>
<td>You can veto editing this cell</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td>This default cell will overflow into neighboring cells, but not if you turn overflow off.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td>This cell is set to span 3 rows and 3 columns</td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>17</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>19</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>21</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>22</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**wxPython:** Cross Platform GUI Toolkit
Widgets

- `wx.gizmos.TreeListCtrl`
Widgets

- `wx.html.HtmlWindow`
Widgets

- **wx.stc.StyledTextCtrl**
  - (wx port of Scintilla)

```python
#!/bin/env python
#---------------------------------------------------------------
# Name: Main.py
# Purpose: Testing lots of stuff, controls, window types, etc.
#
# Author: Robin Dunn
#
# Created: A long time ago, in a galaxy far, far away...
# RCS-ID: $Id: Main.py,v 1.76.2.29 2003/05/23 16:47:49 RD Exp$
# Copyright: (c)1999 by Total Control Software
# Licence: wxWindows license
#---------------------------------------------------------------

import sys, os, time

from wxPython.wx import *

from wxPython.html import wxHtmlWindow

import images
```
Let’s create an application
Let’s create an application

```python
import wx

class App(wx.App):
    def OnInit(self):
        frame = wx.Frame(parent=None, title='Bare Frame')
        frame.Show()
        return True

app = App()
app.MainLoop()
```
Let’s create an application
Event handling
import wx

class MyFrame(wx.Frame):
    def __init__(self, parent, title):
        wx.Frame.__init__(self, parent, -1, title,
        pos=(150, 150), size=(350, 200))

        menuBar = wx.MenuBar()
        menu = wx.Menu()
        menu.Append(wx.ID_EXIT, "E&xit\tAlt-X",
                   "Exit this simple sample")

        self.Bind(wx.EVT_MENU, self.OnTimeToClose,
                   id=wx.ID_EXIT)

        menuBar.Append(menu, "&File")
        self.SetMenuBar(menuBar)
        self.CreateStatusBar()
Simple sample

```python
panel = wx.Panel(self)

text = wx.StaticText(panel, -1, "Hello World!")
text.SetFont(wx.Font(14, wx.SWISS, wx.NORMAL, wx.BOLD))

btn = wx.Button(panel, -1, "Close")
funbtn = wx.Button(panel, -1, "Just for fun...")

self.Bind(wx.EVT_BUTTON, self.OnTimeToClose, btn)
self.Bind(wx.EVT_BUTTON, self.OnFunButton, funbtn)

sizer = wx.BoxSizer(wx.VERTICAL)
sizer.Add(text, 0, wx.ALL, 10)
sizer.Add(btn, 0, wx.ALL, 10)
sizer.Add(funbtn, 0, wx.ALL, 10)
panel.SetSizer(sizer)
panel.Layout()
```
def OnTimeToClose(self, evt):
    self.Close()

def OnFunButton(self, evt):
    print "Having fun yet?"

class MyApp(wx.App):
    def OnInit(self):
        frame = MyFrame(None, "Simple wxPython App")
        frame.Show(True)
        self.SetTopWindow(frame)
        return True

app = MyApp(True)
app.MainLoop()
Simple sample

Hello World!

Close

Just for fun...
For more information

- Join the wxPython-users mail list by sending a message to wxPython-users-subscribe@lists.wxwidgets.org
- Slides of this presentation are available at: http://wxPython.org/OSCON2004/nutshell/
Questions?