CS 683 Emerging Technologies
Fall Semester, 2005
Doc 13 Python For Series 60 p1
Contents

References

Python for Series 60 Platform API Reference

Programming with Python for Series 60 Platform

Useful References

CS457/546 - Information for Python Programming on our Nokia Devices,
http://www.csd.uwo.ca/courses/CS457a/python/index.html

Matt Croydon PythonForSeries60 Wiki,
http://www.postneo.com/postwiki/moin.cgi/PythonForSeries60

Nokia Python for Series 60,
http://www.forum.nokia.com/python

Nokia Python for Series 60 Forum,
http://discussion.forum.nokia.com/forum/forumdisplay.php?forumid=102

Documentation

Python for Series 60 Platform API Reference

Programming with Python for Series 60 Platform

Both are pdf files

Available at
http://www.forum.nokia.com/main/0,,034-821,00.html

Go to bottom of page & click on Python for Series 60 Documentation bundle

Short Installation Instructions
Cell Phone Emulator on Windows

Step 1. Download the following

http://www.csd.uwo.ca/courses/CS457a/tutorial/Symbian_python_tutorial.zip

Step 2 Follow the instructions for installing in the above under Environment

This is the preferred solution if it works

The emulator provides a cell phone interface

Python only works when emulator runs in debug mode

wxPython for Windows, Linux & Mac OS X

Step 1 Install wxPython

Download at: http://wxpython.org/

Download and install version with unicode support

Prebuilt binaries work fine

Follow the instructions with the wxPython download

Step 2 Install Series 60 Compatibility Library

Download at: http://pdis.hiit.fi/pdis/download/

Follow the instructions with the download

This allows code written for Python Series 60 to run using wxPython widgets

- Does not look like applications on cell phone
- Not all features will work

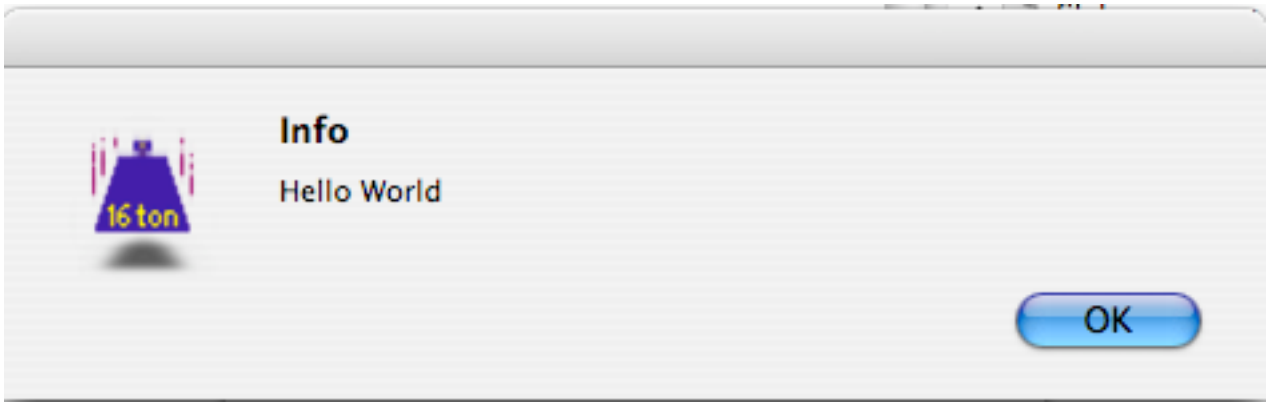
Supports appuifw, e32, key_codes

Hello World Example

```
import appuifw

appuifw.note(u'Hello World', 'info')
```

wxPython Version



Writing Applications for Cell Phones
Technical constraints

- Limited screen space
- Limited storage space
- Limited memory
- Limited power
- Slow processor
- Restricted user input
- Network connection questionable
- Feature set of phones vary
- Memory leaks not allowed

Business Constraints

- Most phones are not programmable
- Multiple incompatible OS/programming systems
- How much are people willing to pay for cell phone program
- Do people see a need for 3'rd party applications
- Main sellers are games and ring-tones

Issues to Consider

- Language
- Features
- Security
- Distribution
- Phones/providers supported

Some Development Platforms

- Qualcomm Brew
  C++, claim some Java support

- Java J2ME
  Support from multiple venders

  Abstraction can limit access to phone features

- WinCE
  Multiple devices

  Multiple development systems

- Symbian OS
  Primarily Nokia phones

- Linux

Symbian

http://www.symbian.com/

Produces OS for advanced cell phones

1200 employees

Owned by:

- Ericsson (15.6%)
- Nokia (47.9%)
- Panasonic (10.5%)
- Samsung (4.5%)
- Siemens (8.4%)
- Sony Ericsson (13.1%)

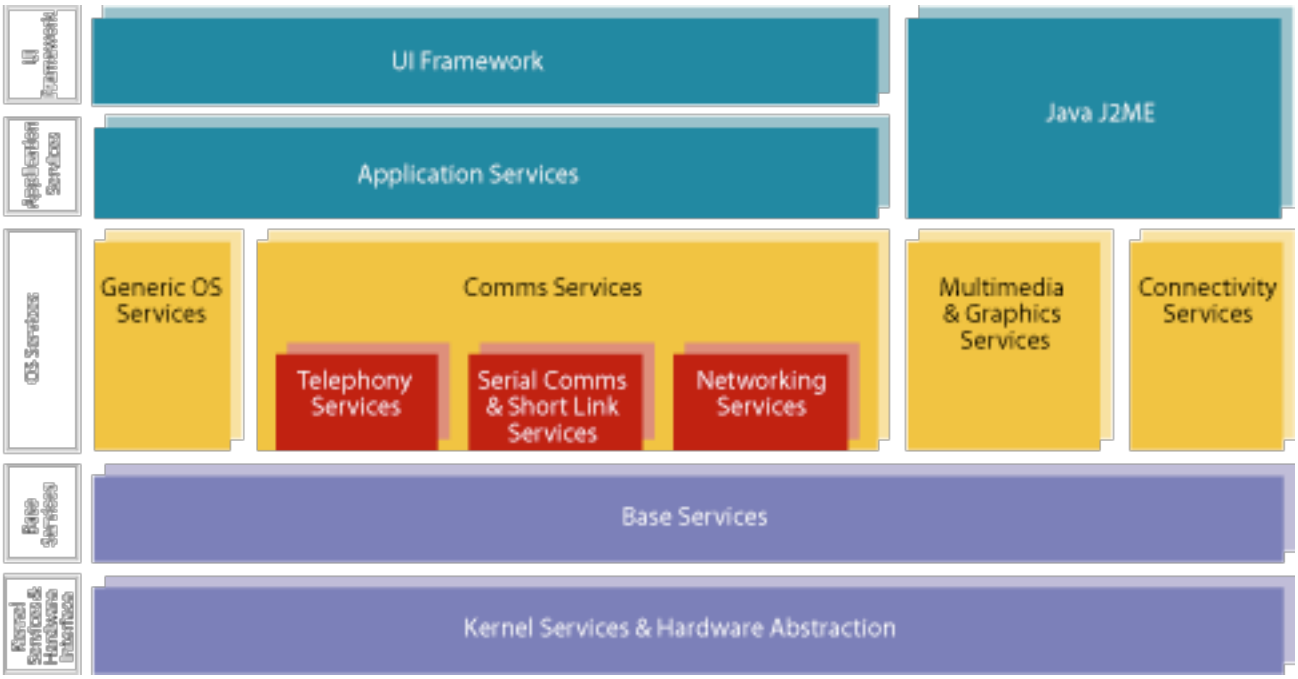
Licensees

Arima	LG Electronics	Panasonic
Ben Q	Motorola	Samsung
Fujitsu	Mitsubishi	Sharp
Lenovo	Nokia	Sony Ericsson

40 million Symbian phones at end of Q1 2005

14.5 million Symbian phones shipped first half of 2005

Symbian OS

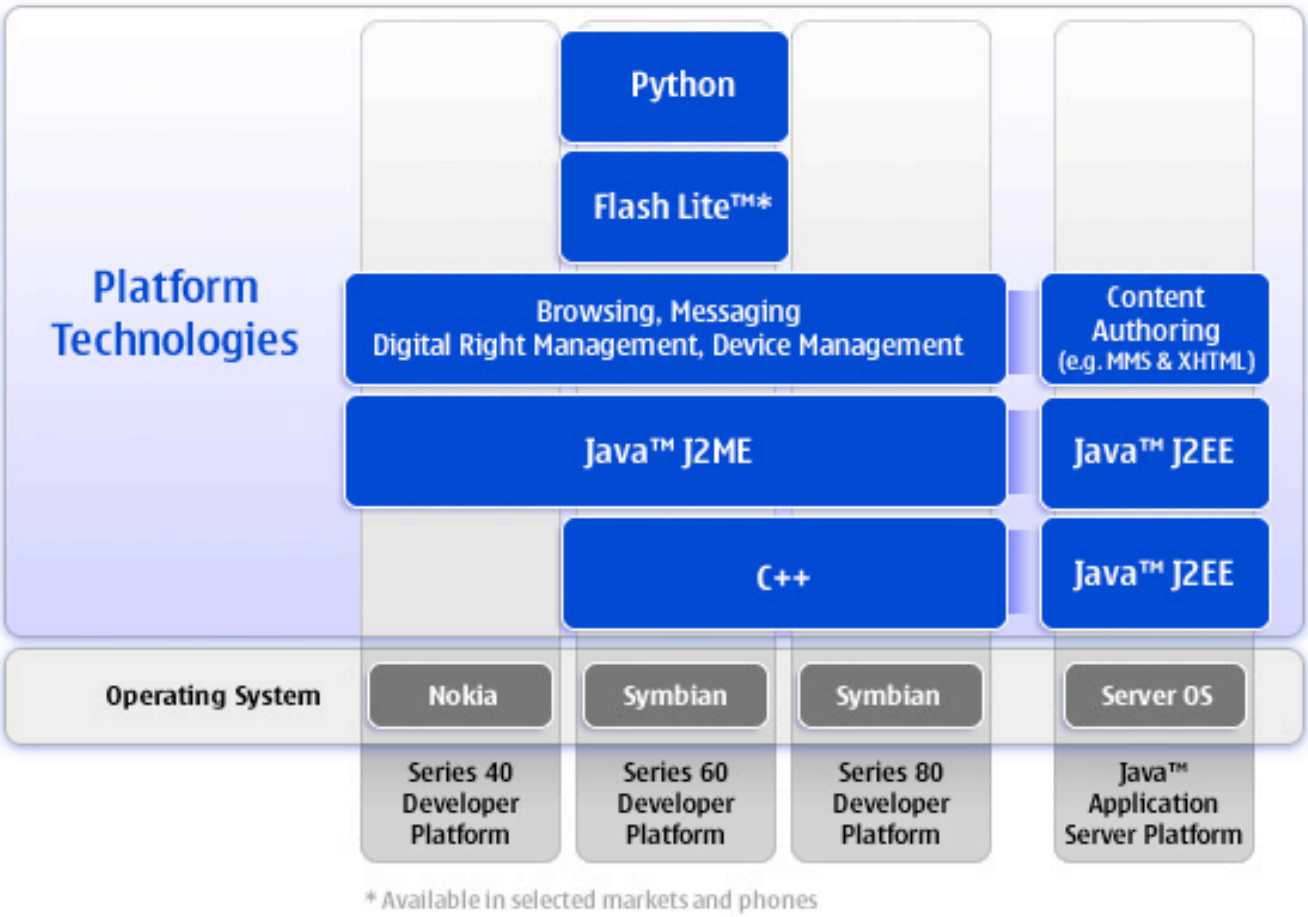


Version 9.0 is real-time, previous versions not real-time

Supports

- GSM, GPRS, EDGE, CDMA
- HTTP, TCP/IP, POP3, SMTP, WAP 2.0
- HTPS, SSL, TLS
- SMS, EMS, MMS
- Cameras,OpenGL
- Contains two DBMS (database) implementations
- multi-tasking
- Version 9.0 is real-time, previous versions not real-time
- Unicode 3.0

Nokia Development Platforms



Nokia Development Platforms

Series 40

- Java only
- 70 phone models
- ~300 million shipped

Series 60

- Java, C++, Python, Visual Basic, C#
- 20 phone models
- ~20 million shipped

Series 80

- Java, C++, Visual Basic, C#
- 2 phone models (Communicator series)

CDMA

- Based on Series 40
- Java only

Series 60 Python Overview

- appuifw
  application user interface framework
- graphics
- e32
  OS related utilities
- e32db
  database access with limited SQL
- e32dbm
  access Symbian RDBMS
- messaging (SMS)
- location - GMS location
- sysinfo
- camera
- audio
  record & play
- telephone
- calendar
- contacts
- Extensions to standard libraries
  - thread
  - socket - added Bluetooth support

appuifw Module

Interface to the Series 60 UI application framework

- Module Functions
- Application
- Forms
- Text
- Listbox
- Icon
- Content_handler
- Canvas

appuifw Functions

```
import appuifw

userInput = 'Hi mom'
while userInput:
    userInput = appuifw.query(u'Enter a string', 'text', userInput)
    if userInput == None:
        break
    appuifw.note(u'You typed: ' + userInput, 'info')

appuifw.note(u'Multi-query', 'info')
answer1, answer2 = appuifw.multi_query(u'A', u'B')

appuifw.note(u'popup_menu', 'info')
selectedIndex = appuifw.popup_menu([u'a', u'b', u'c'])
selectedIndex = appuifw.popup_menu([u'a', u'b', u'c'],\
     u'Select one')

appuifw.note(u'multi_selection_list', 'info')
selectedIndex = appuifw.multi_selection_list([u'a', u'b', u'c', u'd'], \
    style='checkbox', search_field=1)
```

First Application

```
import appuifw
import e32

old_title = appuifw.app.title
appuifw.app.title = u"Hello World"

class HelloWorld:
    def __init__(self):
        self.lock = e32.Ao_lock()
        appuifw.app.exit_key_handler = self.exit_key_handler

    def run(self):
        self.lock.wait()
        self.close()

    def exit_key_handler(self):
        self.lock.signal()

    def close(self):
        appuifw.app.exit_key_handler = None

myApp = HelloWorld()
myApp.run()
appuifw.app.title = old_title
appuifw.menu = None
```