

# **CS 683 Emerging Technologies**

## **Fall Semester, 2005**

### **Doc 16 Python for Series 60 p4**

### **Contents**

Data.....	3
e32dbm.....	4
e32db.....	5
Contacts.....	6
Networking.....	8
SMS.....	9
Phone Calls.....	10
TCP/IP.....	11
Why Python on Cell Phone.....	12

Copyright ©, All rights reserved. 2005 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (<http://www.opencontent.org/opl.shtml>) license defines the copyright on this document.

## **References**

Python for Series 60 Platform API Reference, version 1.1.5

Programming with Python Series 60 Platform, version 1.1.5

## **Data**

- e32db
- e32dbm
- Contacts
- Calendar

## e32dbm

Simple Python dictionary

Very easy tuse

```
import e32dbm

simple = e32dbm.open(u"c:\\simple.db", 'c')
simple['cat'] = 'mouse'
simple['dog'] = 'sam' #values & keys must be strings
simple.sync()
for x in simple:
    print x
print simple['cat']
simple.close()
```

### Methods & Details

See chapter 9 (pp 42-43) of API Reference for Python Series 60

Methods are subset of mapping methods

See <http://docs.python.org/lib/typesmapping.html> for explanation of methods

## e32db

Uses subset of SQL

Compatible with C++ code

Used by e32dbm

```
import e32db
db = e32db.Dbms()
db_name = u"c:\\Books.db"
try:
    db.open(db_name)
except:
    db.create(db_name)
    db.open(db_name)
    create_books = u"CREATE TABLE books (title VARCHAR,
        author VARCHAR)"
    db.execute(create_books)

add_book = u"INSERT INTO books (title, author) VALUES
    ('%s', '%s')"%('A Cat in the Hat', 'Dr Seuss')
db.execute(add_book)
all_books = e32db.Db_view()
all_books.prepare(db, u"SELECT * from books")
all_books.first_line()

for k in range(all_books.count_line()):
    all_books.get_line()
    print all_books.col(1)
    print all_books.col(2)
    all_books.next_line()
db.close()
```

See

- Chapter 8 Python for Series 60 Platform API Reference
- Chapter 9 Programming with Python

# Contacts

## Classes

ContactDb - database of contact objects

Contact - represents a contact, contains fields

### Possible Contact Fields

city	first_name	postal_address
company_name	job_title	postal_code
country	last_name	state
date	mobile_number	street_address
dtmf_string	note	url
email_address	pager_number	video_number
extended_address	phone_number	wvid
fax_number	po_box	

ContactField, contains attributes

- label
- value
- type
- location (none, work, home)
- schema

## Contact Example

```
import contacts

contacts = contacts.open()
new_person = contacts.add_contact()
new_person.add_field('first_name', value='Roger')
new_person.add_field('last_name', value='Whitney')
new_person.add_field('mobile_number', value='1111111111')
new_person.commit()

for k in contacts.keys():
    print contacts[k].find('first_name')[0].value
matches = contacts.find('Whitney')
print matches[0].find('last_name')[0].value
```

contacts.find( ) return list of contact objects

contactObject.find('field\_name') returns list of contact fields

## **Networking**

- SMS text messages
- Telephone
- TCP/IP
- Bluetooth



## SMS

```
import messaging
```

```
messaging.sms_send('6165831978', u'Hello')
```

### Simulator

- Does not send message
- Records the message  
C:\Symbian\6.1\Series60\Epoc32\Winds\c\system\Mail

## Phone Calls

```
import telephone  
telephone.dial(u'6191111111')
```

Has no effect in simulator

## Location

```
import location  
  
country_code, network_code, area_code, cell_id = \  
    location.gsm_location()  
print country_code, network_code, area_code, cell_id
```

Fake data returned in simulator

## TCP/IP

Uses standard Python networking modules

- socket

<http://docs.python.org/lib/module-socket.html>

- urllib

<http://docs.python.org/lib/module-urllib.html>

### socket

low level access to network

Bluetooth support added

### urllib

```
import urllib
```

```
page = urllib.urlopen('http://www.eli.sdsu.edu/index.html')  
print page.read()
```

Simulator does not seem to access network

## **Why Python on Cell Phone Hello World C++ version Files**

- bld.inf
- helloworld.h
- helloworld.hrh
- helloworld.mmp
- helloworld.rss
- helloworld\_application.cpp
- helloworld\_appui.cpp
- helloworld\_appview.cpp
- helloworld\_document.cpp
- helloworld\_main.cpp

## Hello World Java J2ME Version

```
import javax.microedition.midlet.*;
import javax.microedition.lcdui.*;

public class MySample extends MIDlet {

    public MySample() { }

    public void startApp() {
        Form form = new Form( "First Program" );
        form.append("Hello World" );
        Display.getDisplay(this).setCurrent( form ); }

    public void pauseApp() { }

    public void destroyApp( boolean unconditional ) { }
}
```

## **Hello World Python Version**

```
import appuifw
```

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