

Calling C functions from within Python using Mac OS X

Christian Stigen Larsen

March 28, 2006

1 Introduction

Here's a small tutorial on how to call your C functions from within Python.

2 Your C program

Let's make some simple functions from within C. We'll call the file *myModule.c*.

```
#include <Python.h>

// Function to be called from Python
static PyObject* py_myFunction(PyObject* self, PyObject* args) {
    char* s = "Hello from C!";
    return Py_BuildValue("s", s);
}

// Another function to be called from Python
static PyObject* py_myOtherFunction(PyObject* self, PyObject* args) {
    double x, y;
    PyArg_ParseTuple(args, "dd", &x, &y);
    return Py_BuildValue("d", x*y);
}

// Bind Python function names to our C functions
static PyMethodDef myModule_methods[] = {
    {"myFunction", py_myFunction, METH_VARARGS},
    {"myOtherFunction", py_myOtherFunction, METH_VARARGS},
    {NULL, NULL}
};

// Python calls this to let us initialize our module
void initletmyModule() {
    (void) Py_InitModule("myModule", myModule_methods);
}
```

3 Compilation

Compiling dynamic libraries on Mac OS X is different from the usual *gcc -shared* you might be used to:

```
gcc -dynamiclib -I/usr/include/python2.3/ -lpython2.3 -o myModule.dylib myModule.c
```

Now you have to do something awkward; rename `myModule.dylib` to `myModule.so` so that Python will find the correct file (this is a bug in Python, it should've been fixed, but that's as far as I know):

```
mv myModule.dylib myModule.so
```

If you are using a system that supports *gcc -shared* you can do this:

```
gcc -shared -I/usr/include/python2.3/ -lpython2.3 -o myModule.so myModule.c
```

4 Calling your functions from Python

Here's a simple program in Python to call your functions:

```
from myModule import *

print "Result from myFunction():", myFunction(), "."
print "Result from myOtherFunction():", myOtherFunction(4.0, 5.0), "."
```

The output is this:

```
Result from myFunction(): Hello from C! .
Result from myOtherFunction(): 20.0 .
```

If you are going to make bigger libraries available from Python I suggest you check out SWIG on the web.