

# Re: Building things with setup.py

---

*Source:* <http://coding.derkeiler.com/Archive/Python/comp.lang.python/2006-09/msg03385.html>

---

- *From:* "Martin v. Löwis" <[martin@xxxxxxxxxxx](mailto:martin@xxxxxxxxxxx)>
  - *Date:* Sat, 23 Sep 2006 08:02:40 +0200
- 

James Stroud schrieb:

I think I would like to know how to avoid or correct these sort of issues in the future, which seem to be limited, for me at least, to scipy and numpy, with the possible exception of MySQLdb and its dependency on zlib. Ideally, I would like to understand exactly what causes these errors in addition to knowing what I can do to correct them in the future.

Let's take a specific failure, namely the line

```
/auto_nfs/data10/users/jstroud/Programs/bin/g77
-L/data10/users/jstroud/Programs/lib
-L/usr/lib/gcc-lib/i386-redhat-linux/3.2.3
build/temp.linux-i686-2.5/numpy/core/blasdot/_dotblas.o
-L/data10/users/jstroud/Programs/lib -lblas -lg2c -o
build/lib.linux-i686-2.5/numpy/core/_dotblas.so
```

This gives errors like

```
build/temp.linux-i686-2.5/numpy/core/blasdot/_dotblas.o(.text+0x758):numpy/core/blasdot/_dotblas.c:226:
undefined reference to `PyTuple_New'
```

That's not too surprising: this line tries to link the input \*as an executable program\*, despite calling the output `_dotblas.so`. In an executable program, all symbols need to be defined; that's why it complains about so many missing symbols (including `MAIN__` – which should never be missing in a library). Even though adding `python2.5.a` to the linker link makes these symbols appear, the result still won't work, as you can't use an executable program file as if it were a shared library.

Now, compare this to a succeeding build of a C extension module,

```
gcc -pthread -shared -L/data10/users/jstroud/Programs/lib
-L/usr/lib/gcc-lib/i386-redhat-linux/3.2.3
```

## Re: Building things with setup.py

```
-I/data10/users/jstroud/Programs/include  
-I/data10/users/jstroud/Programs/qt/include -I/usr/include  
-I/data10/users/jstroud/Programs/include/freetype2/freetype  
build/temp.linux-i686-2.5/build/src.linux-i686-2.5/numpy/core/src/umathmodule.o  
-lm -o build/lib.linux-i686-2.5/numpy/core/umath.so
```

Notice that this passes `-shared` to the compiler, requesting construction of a shared library. This is the problem with the `g77` linker line; try invoking

```
/auto_nfs/data10/users/jstroud/Programs/bin/g77 -shared  
-L/data10/users/jstroud/Programs/lib  
-L/usr/lib/gcc-lib/i386-redhat-linux/3.2.3  
build/temp.linux-i686-2.5/numpy/core/blasdot/_dotblas.o  
-L/data10/users/jstroud/Programs/lib -lblas -lg2c -o  
build/lib.linux-i686-2.5/numpy/core/_dotblas.so
```

manually (whether or not `-pthread` is also necessary or supported for `g77`, I don't know). This should at least make the complaints about missing symbols go away; you should then see whether the resulting module can be imported in Python.

If that solves the problem, the question is why the `-shared` option isn't passed automatically; your setting `LDFLAGS` might indeed be a cause.

Regards,  
Martin

.