

# embedding python- a new approach to specifying program options

## Embedding python in a C++ application

I've developed a simple interface package, taking advantage of the fact that all applications built with our tools can link and run against the Python interpreter, to allow a seamless invocation of Python code from C++. Jim Chiang has provided invaluable help and suggested extensions.

My purpose was to make it easy to specify configuration data for applications, avoiding wiring in constants, or writing elaborate parsing difficult to justify for small, or one-off applications. But the capability extends far beyond that.

The very small interface packages is "embed\_python". It is currently in ScienceTools LATEST, and will migrate to the release soon. Non-ScienceTool users can check it out and build it.

Here is a complete package that runs a test program, named "test\_python", in only three files. The requirements file:

```
package test_python

use embed_python *

application runit main.cxx
```

The file src/main.cxx:

```
#include "embed_python/Module.h"

#include <iostream>
#include <string>

void main(){

    const char * here (::getenv("TEST_PYTHONROOT"));
    embed_python::Module runinfo( std::string(here)+"/python", "setup");

    std::string hello;
    runinfo.getValue("hello", hello);

    std::cout << hello << std::endl;

}
```

and the file "python/setup.py":

```
# simple script :-)

hello = "hello, world"

print "I've set you up!"
```

Some advantages are:

- Syntax for setting simple variables very readable (compared with, say XML)
- Flexible, extensible
- Easy to add comments to the data file, generate output when processed
- Possible to make computations, guarantee restrictions
- Messages about syntax errors are automatic and informative
- Setup file can be tested with stand-alone python

Of course, if XML is important, and the interface is simple, it is much easier to parse XML in python than C++.

It is now in use by the IRF code.

Additional features that are supported, and easy to use, are loading lists and calling functions.