

[Creating a Python module using Python 3.2 and MinGW]

Download and install Python 3.2 [Add the [installation_path](#) of Python to the PATH environment variable]

Download and install MinGW [Also add the [installation_path\MinGW\bin](#) directory to the PATH environment variable]

Download the archive [PythonModule.zip] that contains the following:

Build_example – An example of the directory that is created when the create_module.bat is run

Create_module.bat – A simple script to create the module.

Main.cpp – The C++ code that will be stored in the module

Setup.py – A small Python script used to create the new module

[Step 1] Run the create_module.bat file that will create a new Python module:

[build\lib.win32-3.2\mymodule.pyd](#)

[Step 2] Open Python in a command line window in the build\lib.win32-3.2 directory and run the following commands to load and use the newly created module:

```
>>> Import mymodule
>>> mymodule.system("ipconfig")
```

This should display the normal ipconfig command results in Windows.

(This guide may not be complete)

To fix the Visual Studio 2010 linker error:

[Step 1] Navigate to the installation_path\include directory and open the pyconfig.h file

[Step 2] Change the following code found within the pyconfig.h file:

```
#ifdef _DEBUG
#    define Py_DEBUG
#endif
```

To:

```
/*
#ifdef _DEBUG
#    define Py_DEBUG
#endif
*/
```

[Step 3] Also change the following code found within the pyconfig.h file:

```
pragma comment(lib, "python32_d.lib")
```

To:

```
pragma comment(lib, "python32.lib")
```