Setting Up PySZG on Mac OS X 10.5

William Davis

Thursday, July 10, 2008

Introduction

This report simply details the method I used to get PySZG working with our current Syzygy version on Mac OS X 10.5. I'm not claiming that this is a great way to do it - but as of today it's the only way I've found. This setup uses Python 2.3 and SWIG 1.3.21, with the Wii Syzygy 1.1 that has been patched for the Mac OSX 10.5 Darwin kernel. There seems to be some path issues which require that the core Syzygy dynamic libraries (*.dylib) be located in the same directory as the `python2.3` executable, though there is probably a simple fix to this.

What You Need

This section assumes only a few things - that you are using Mac 10.5 in Altgeld 102, you have the patched version of Syzygy installed (all machines we use in here currently do), and Python 2.3 installed as well (as with Syzygy, Python 2.3 is also installed on all machines in here) - however, the current installation that Mac defaults to is Python 2.5. We will circumvent this without destroying the 2.5 installation. The basic outline is as follows:

- Re-patch the `arDataUtilities` source code in Syzygy
- Re-install the newly patched Syzygy
- Get and install SWIG 1.3.21
- Bind the Python and Syzygy libraries using SWIG
- Clean things up
- Enjoy the fruits of your labor!

Repatching Syzygy

The 1.1 version of Syzygy has a recent bug in `arDataUtilities.cpp` and `arDataUtilities.h` that prevents PySZG from binding correctly. It's a simple fix, and hopefully soon it should have proliferated among the lab machines so this part can be skipped. Many thanks to Jim Crowell for pointing out the bug.

To fix it, change directories to the Syzygy source code:

```
$ cd /sw/syzygy/wii/szg/src/language
```

We're going to edit the source files, but first make backups:

```
$ cp arDataUtilities.cpp arDataUtilities.cpp.back
$ cp arDataUtilities.h arDataUtilities.h.back
```

Hopefully you won't have to use them, but just to be safe. Use whatever editor you like to edit the `arDataUtilities` files - I will use `vim` to edit:

```
$ vim arDataUtilities.cpp
```

We need to find the following block of code:

```
static string __version("");

string ar_versionString() {
  if (__version == "") {
    ostringstream os;
    os << SZG_MAJOR_VERSION << "."
       << SZG_MINOR_VERSION << "."
```

```
      << SZG_PATCH_VERSION;
    __version = os.str();
  }
  return __version;
}
```

We see that this block of code comes shortly after a compiler directive:

```
#ifdef AR_USE_WIN_32
```

We need to move the above block of code to outside this compiler directive, so move it after the line

```
#endif
```

but make sure it is the one that immediately follows the `#ifdef` statement - then save and quit.
Now for `arDataUtilities.h`:

```
$ vim arDataUtilities.h
```

Find the line that says:

```
string SZG_CALL ar_versionString();
```

and change it to:

```
SZG_CALL string ar_versionString();
```

Save the changes and quit. If anything goes terribly wrong, you can always revert back to your backup files and start over.

Re-installing Your Patched Syzygy

To reinstall the patched Syzygy, change directories to the build directory:

```
$ cd /sw/syzygy/wii/szg/build/
```

Make it clean first:

```
$ make demo-clean
$ make clean
```

and then build it again:

```
$ make demo
```

and that should do it.

Get and Install SWIG 1.3.21

To get SWIG 1.3.21, go to this link:

```
http://sourceforge.net/project/downloading.php?groupname=swig&
   filename=swig-1.3.21.tar.gz&use_mirror=internap
```

or Google "SWIG 1.3.21" and find the Sourceforge link (should be one of the top results). Put it wherever you like and unpack it:

```
$ gunzip swig-1.3.21.tar.gz
$ tar -xf swig-1.3.21.tar
$ cd SWIG-1.3.21
```

Now configure SWIG for installation and install it:

```
$ ./configure
$ make
$ make install
```

Again, should anything go wrong, you can uninstall it by typing:

```
$ make uninstall
$ make clean
```

and that should cover it. SWIG is probably smart enough to put itself in all the right places, but you can check it by going to an arbitrary place in your filesystem and typing:

```
$ swig -version
```

to check that the SWIG binaries are on your PATH. If not, you may have to add your bin to the path:

```
$ export PATH=$PATH:/usr/local/bin
```

It may be helpful to add that line to your .bash_profile, as is described in the **Cleaning Up** section.

Bind the Python and Syzygy Libraries

While you have just installed SWIG, you won't actually have to explicitly call it - this is built in to the PySZG makefiles. Before you make PySZG, however, you need to set a few environment variables. First, get the szguser.profile from dent. Connect to dent from the Finder by hitting the Apple-Command (Flower) + K. A dialog will pop up, and in the top text field type:

```
afp://dent
```

and click Connect. Enter in your user name and password in the subsequent dialog and click Connect once again, and from the list that comes up select netgroups and click OK. Navigate into the illimath directory, and copy the file szguser.profile to your home directory, /Users/<Your_User_Name>/

Now from the command line, navigate to your home directory and source the user profile:

```
$ cd ~
$ source szguser.profile
```

which should set up a few of your environment variables, namely $SZGHOME and $SZGBIN. We still need a couple more however. Issue the following commands from the terminal:

```
$ export SZG_PYINCLUDE=/sw/include/python2.3/
$ export SZG_PYLIB=
$ export PATH=$PATH:$SZG_PYINCLUDE
$ ln -s python2.3 python
```

The first tells Syzygy where the header Python.h is. The second doesn't actually need to be done, but we do it just in case SZG_PYLIB has been previously set. The third line will put the python2.3 executable on your path, so it can be executed from anywhere. The final line is optional - it will override the current Python installation on the machine, by linking the Python 2.3 executable to the word python. If you still wish to use Python 2.5, I would suggest skipping this line.

Now change directory into the PySZG directory, now known as python/ within the Syzygy directory tree, and make:

```
$ cd $SZGHOME/python/
$ make
```

If it compiles, all is well so far. If not, this installation method will not work and there is something wrong. But we're still not quite ready yet - there are a few more things to be done. We must add the /sw/bin/ to the PATH, link the libraries there, and set the PYTHONPATH:

```
$ export PATH=$PATH:/sw/bin/
$ ln -s $SZGBIN/* /sw/bin/
$ export PYTHONPATH=$SZGBIN
```

The second line is very messy - I'm not sure precisely which files it needs to run - it may just need the dynamic libraries, so you could first try the following:

```
$ ln -s $SZGBIN/*.dylib /sw/bin
```

But nonetheless, it should work if you dump everything in there. PySZG should now be ready - you can test it by executing the following:

```
$ python2.3
>>> from PySZG import *
>>> g = arGraphicsDatabase()
>>> g.printStructure()
```

If the above executes without errors, your installation was a success. To exit the Python interpreter, just type `Ctrl+D`. You can further test it by trying the demos:

```
$ cd $SZGHOME/python/demo/blobby/
$ python2.3 blobby.py
```

If this does not work, again, sorry, something is wrong.

Cleaning Up

This method leaves much to be desired: it is complicated to set up, it is not modular at all, and the paths have to be reset each time a new bash terminal is started. We can at least alleviate the last one of these slightly by building it into a shell script.

If you want this to be readily available each time you open a new terminal, you can edit/create a file called `.bash_profile` in your home directory (`/Users/<Your_User_Name>/`). Simply add the following to the file[1]:

```
$ export SZGHOME=/sw/syzygy/wii/szg/
$ export SZGBIN=/Users/<Your_User_Name>/szgbin/
$ export SZG_PYINCLUDE=/sw/include/python2.3/
$ export SZG_PYLIB=
$ export PYTHONPATH=$SZGBIN
```

Or, if you don't wish to edit your `.bash_profile`, you can add the last three of those to the `szguser.profile` and `source` it every time you open a new shell.

And that should do it! This is definitely an imperfect process, and I'm open to improvement on it.

[1] Note that if PySZG was not set up in your account, you should replace Your User Name with the name of the account that PySZG was set up under, in the case of euclid, this user is reulab