Installing GDAL on a Mac

The only way I have experience installing GDAL on a Mac is using the frameworks from http://www.kyngchaos.com/software:frameworks, and I've only done it on Leopard. It looks like there are frameworks built for Tiger as well, and build scripts available for Panther. This is how I did it for Leopard, and Tiger should be pretty much the same. If you need to install it on Panther and are lost, I can try to help.

There will also be a few differences from how we do things in class. You will need to use the NumPy module rather than Numeric for processing large data arrays (they have a lot in common, so you shouldn't find it too different) and you will have to import the GDAL modules slightly differently (see the end of this document).

It is important to install the frameworks in the correct order. See http://www.kyngchaos.com/software:support:install for a list of all frameworks available and the order in which they must be installed. If you want a simple GIS on your Mac (which would be useful, so you could open your data sets to see if your scripts actually worked), then you might want to install QGIS as well.

Installing Python on Tiger

If you're using Tiger you need to install Python 2.5. Do not do this for Leopard because it comes with the correct version of Python already.

1. Install Python 2.5 from http://www.python.org/download/releases/2.5.4/. It looks like this is python-2.5.4-macosx.dmg.
2. Install NumPy 1.x for Python 2.5 from http://pythonmac.org/packages/py25-fat/. It looks like this is numpy-1.0.4-py2.5-macosx10.4-2007-11-07.dmg.

Installing the required frameworks on both Leopard and Tiger

All of these steps are required for both Leopard and Tiger. I've included everything you need for both GDAL and QGIS (I hope!).

To install a framework:

1. Download the framework file from http://www.kyngchaos.com/software:frameworks.
2. Double-click the file downloaded in step 1 (it should be a .dmg file) – this will mount the disk image.
3. You should see an .mpkg file in the mounted image. Double click this and follow the directions.

To install everything required for GDAL and QGIS (without GRASS support):

1. Install the latest UnixImageIO framework – UnixImageIO framework v1.0.27 for me.
2. Install the latest SQLite3 framework – SQLite3 framework v3.6.10-1 for me.
3. Install the latest PROJ framework – PROJ framework v4.6.1-3 for me.
4. Install the latest GEOS framework – GEOS framework v3.0.3-1 for me.

5. Install the GDAL framework – GDAL framework v1.6.0-5 for me. You can also install the ECW and MrSID plugins if you want, but they won't be needed for this class.
6. Install QGIS.
   a. Download the correct version of QGIS from http://www.kyngchaos.com/software:qgis – Qgis 1.0.0-2 for me.
   b. Double-click the file downloaded in the previous step and once the disk image is mounted, drag Qgis.app to your Applications folder.

Running Python in Terminal

You should be able to run Python interactively (like in the interactive window in PythonWin) from a Terminal window just by typing `python`. You can run scripts by typing

```
python <script_name> [<args>]
```

You must provide the full path to the script unless it is in the current directory. For example, this doesn't work because I didn't provide the full path to my script:

```
~$ python rasterinfo.py bear_lake.img
/System/Library/Frameworks/Python.framework/Versions/2.5/Resources/Python.app/Contents
/MacOS/Python: can't open file 'rasterinfo.py': [Errno 2] No such file or directory
```

It still doesn't work because I didn't provide the full path to the input file:

```
~$ python /Volumes/MacA/Data/Classes/Python/rasterinfo.py bear_lake.img
ERROR 4: `bear_lake.img' does not exist in the file system,
and is not recognised as a supported dataset name.
```

Now it works:

```
~$ python /Volumes/MacA/Data/Classes/Python/rasterinfo.py
/Volumes/MacA/Data/Imagery/ASTER/bear_lake.img
/Volumes/MacA/Data/Imagery/ASTER/bear_lake.img has 5033 rows, 5665 columns, and 3
bands
```

I can also change into the directory of the script and run it without the full script path:

```
~$ cd /Volumes/MacA/Data/Classes/Python
/Volumes/MacA/Data/Classes/Python$ python rasterinfo.py
/Volumes/MacA/Data/Imagery/ASTER/bear_lake.img
/Volumes/MacA/Data/Imagery/ASTER/bear_lake.img has 5033 rows, 5665 columns, and 3
bands
```

Running Python from an editor

I use jEdit (http://www.jedit.org/) for writing scripts on my Mac. You can use any editor you like, as long as the files are saved as plain text. I would suggest using one that will let you configure it to run Python from inside of it, however. This functionality exists in jEdit – it's under Macros | Misc | Run Script (although it's worth noting that there cannot be spaces in the path and filename of your script). Here are some basic configuration changes I made to my jEdit installation so that it fit my workflow better:

1. My modifier keys didn't seem to be mapped correctly (for example, command didn't seem to really be command). Here is the note I saved for myself about fixing the problem:

To get the keys to work right for keyboard shortcuts on a Mac, edit `/Applications/jEdit 4.2/startup/startup.bsh`, uncomment and change the order around a bit to get:

```
//KeyEventTranslator.setModifierMapping(InputEvent.CTRL_MASK,
//  InputEvent.ALT_MASK, InputEvent.META_MASK,
//  InputEvent.SHIFT_MASK);

/* ... and this the MacOS default: */
KeyEventTranslator.setModifierMapping(
  InputEvent.META_MASK,  /* == C+ */
    InputEvent.ALT_MASK,   /* == A+ */
    InputEvent.CTRL_MASK,  /* == M+ */
    InputEvent.SHIFT_MASK  /* == S+ */);
```

and uncomment:

```
Debug.ALT_KEY_PRESSED_DISABLED = false;
Debug.ALTERNATIVE_DISPATCHER = false;
```

2. Assign a keyboard shortcut for the Run Script macro, because I *hate* using the menus.
   a. Open the Preferences dialog from the jEdit menu.
   b. Go to the Shortcuts pane.
   c. Choose Macros from the Edit Shortcuts dropdown list at the top.
   d. Scroll down to Run Script and click in the Primary Shortcut field.
   e. Enter your desired shortcut. I used option-r, which displays as A+r in the dialog. It will tell you if your chosen shortcut is already assigned to something else.
   f. Once you're happy with your shortcut, hit OK a few times to get back to the main jEdit window.
3. Another thing that really bothered me was that jEdit had the ability to comment a chunk of code (Edit | Source | Line Comment), but not uncomment it. That was completely unacceptable, in my opinion, so I wrote a script to solve the problem. Here's how you install it:
   a. Get the uncomment.bsh script from the class website and save it in the `/Users/<your_name>/.jedit/macros/` folder. So I ended up with a file called `/Users/chrisg/.jedit/macros/uncomment.bsh`.
   b. Start jEdit (or restart if it's already open) and then you should see a macro called uncomment under the Macros menu.
   c. If I were you, I would set a keyboard shortcut for this macro using the method described in Step 2.

Importing GDAL modules

Import modules like this instead of how we do it in class:

```
from osgeo import gdal
from osgeo import ogr
from osgeo import osr
from osgeo import gdal_array
from osgeo import gdalconst
from osgeo.gdalconst import *
```