

Installing GDAL for Python on Windows

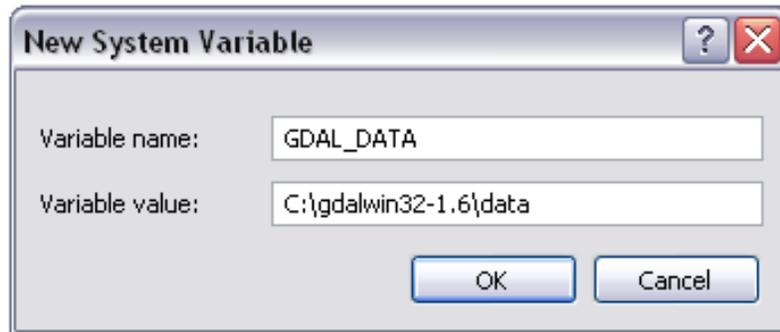
You can install GDAL to work with versions of Python other than the one that comes with FWTools. Some things will be different, however. You will use the NumPy module rather than Numeric for processing large data arrays (they have a lot in common, so you shouldn't find it to be too different) and you will have to import the GDAL modules slightly differently (see the end of this document).

These directions should work for Python versions 2.4 and 2.5. Just make sure you grab things for your Python version if it's different from mine. If you have ArcGIS 9.2 then you should already have Python 2.4. If you have ArcGIS 9.3 then you should have Python 2.5. If you don't have Python already you can get it from <http://www.python.org/download/> (as I write this on 3 February 2009 the latest version of Python you can use GDAL with is 2.5, so don't get a newer one).

1. Make sure you have Python installed.
2. Install the GDAL Windows binaries. You need these before you can install the Python bindings for GDAL.
 - a. Download the latest version of GDAL for Windows from <http://download.osgeo.org/gdal/win32/>. I grabbed gdalwin32exe160.zip from the 1.6 folder.
 - b. Unzip the GDAL archive into a permanent location. I unzipped it right to the C:\ drive, so I ended up with a folder called C:\gdalwin32-1.6.
 - c. Add your new GDAL bin directory (C:\gdalwin32-1.6\bin in my case) to your system Path environment variable.
 - i. Open your Control Panel.
 - ii. Open the System section (under Performance and Maintenance if you're using the Category view).
 - iii. Go to the Advanced tab.
 - iv. Click on the Environment Variables button.
 - v. Find the Path entry in the System variables list, highlight it, and click Edit.
 - vi. Add a semicolon and the full path to your GDAL bin directory to the end of the Variable value textbox. Do not delete anything that already exists in this box!
 - vii. Click OK (just once – you're going to add some more environment variables).



- d. Add a new GDAL_DATA environment variable.
 - i. On the Environment Variables dialog, click the New button under System variables.
 - ii. Set the Variable name to GDAL_DATA.
 - iii. Set the Variable value to the full path to your GDAL data directory (C:\gdalwin32-1.6\data in my case).
 - iv. Click OK. You will need to add more environment variables later, so you can leave the dialog open, or close it until you need it again.



3. Install the PROJ.4 Windows binaries. You need these in order to be able to use projections.
 - a. Download PROJ.4 from <http://download.osgeo.org/proj/>. It looks like the only Windows binary is a few versions behind, but it should be sufficient. It's the file called proj446_win32_bin.zip.
 - b. Unzip the PROJ.4 archive into a permanent location. I unzipped it right to the C:\ drive, so I ended up with a folder called C:\proj.
 - c. Add your new PROJ.4 bin directory (C:\proj\bin in my case) to your system Path environment variable. To do this, follow the directions for part 2c, but add C:\proj\bin instead of C:\gdalwin32-1.6\bin.
 - d. Add a new PROJ_LIB environment variable. To do this, follow the directions for part 2d, but the Variable name should be PROJ_LIB and the Variable value should be the path to your PROJ.4 nad directory (C:\proj\nad in my case). This is the last environment variable you need to add.
 - e. Copy proj.dll from your PROJ.4 bin directory into your GDAL bin directory. In my case, I copied C:\proj\bin\proj.dll to C:\gdalwin32-1.6\bin\proj.dll.
 - f. Create a new directory in your PROJ.4 nad directory and move all of the existing files in that directory into this backup directory (or just delete everything that's in the nad directory, but I'm not sure how safe that is – it might turn out that we need some of those files).
 - g. Grab the Datum shift files from the course website and extract them into your PROJ.4 nad directory.
4. Install the NumPy module for Python. I think this has to be done before installing the Python bindings for GDAL.
 - a. Download the appropriate version of NumPy from http://sourceforge.net/project/showfiles.php?group_id=1369&package_id=175103. Make

sure you get the one for your version of Python! For me, this was numpy-1.2.1-win32-superpack-python2.5.exe.

- b. Install NumPy by running the file you downloaded in the previous step.
5. Install the GDAL Python bindings.
 - a. Download the appropriate version of the GDAL Python bindings from <http://pypi.python.org/pypi/GDAL/>. The downloads are way down at the bottom of the page. Again, make sure you get the one for your version of Python. For me, this was GDAL-1.6.0.win32-py2.5.exe.
 - b. Install the GDAL Python bindings by running the file you downloaded in the previous step.
6. You might have to reboot, but don't do that unless you want to or things aren't working.

There are more details at <http://trac.osgeo.org/gdal/wiki/GdalOgrInPython>.

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 *
```