



Python, GIS and stuff...

A technical blog concerning Python programming and GIS applications

GDAL/OGR, PYTHON

OLD METHOD: Installing GDAL / OGR for Python on Windows

Posted on July 7, 2011December 16, 2017 by StacyR in GDAL/OGR, Python

It is now even easier to install GDAL for Python on Windows – see the new post here: [Installing GDAL / OGR for Python on Windows](#) ! **Do not use the steps below unless you have issues with the above...**

After years of having different GDAL versions on various Windows and Linux setups I have finally found out how to install recent versions of GDAL (and OGR) on Windows; and it is surprisingly easy! *You will need administrator rights on your computer.*

Note that the following situations may cause issues with installing GDAL for Python:

- **Multiple Python installations.** For example: ArcGIS installs a version of Python that is not officially registered with Windows (making it a bit more tricky to install GDAL to it). See step 1 for more information.
- **GDAL may already be installed and required by other software (i.e. PostGIS).** See step 7 for information and workarounds.

Overview:

1. *Determine your desired Python install and version*
2. *Get the GDAL core files*
3. *Get the Python bindings*
4. *Install GDAL*
5. *Edit Environment Variables*

- 6. *Install Python bindings*
- 7. *Issues/workarounds for existing GDAL installations*
- 8. *Common issues*

1. Determine your desired Python install and version

See the post on [Locating Python and adding it to your system Path](#) for information about how to find the Python install you wish to use (you do not need to add it to your system Path to continue with the steps below, but it may be useful).

Once you have figured out which Python you are going to use, open the Python interpreter in your normal way (or double click on its python.exe). It will print version information at the top when it opens, for example: Python **2.7.10** (default, May 23 2015, 09:40:32) [MSC v.1500 **32 bit** (Intel)] on win32

The important information is highlighted above: my version is **Python 2.7** (the additional .10 is not important) and it is **32 bit**.

2. Get the GDAL core files

This site: <http://www.gisinternals.com/release.php> is where you download GDAL. Scroll down to the bottom of the table and click on the relevant link in the 'Downloads' column – the Arch. column either shows if it is 32 (win32) or 64 bit (x64), **select the one corresponding to your Python**.

From the new page download the GDAL Core file: gdal-[version]-[build]-core.msi (e.g. gdal-111-1800-core.msi)

3. Get the Python Bindings

Fortunately the Python bindings are located right here as well, called GDAL - [gdalVersion.architecture]-py[pythonVersion].[exe/msi] (e.g. GDAL-1.11.1.win32-py2.7.msi), just make sure to download the version matching your installed version of Python.

4. Install GDAL

Simply run the GDAL Core installer downloaded in step 2, and note the install path if you change it.

5. Edit Environmental Variables

WARNING: Don't make any mistakes here (like deleting things), or your system might not work...

To bring up the Environment Variables dialogue:

- o open Control Panel
- o go into System
- o click on Advanced system settings on the left
- o then click Environment Variables at the bottom

First you need to identify if there will be any conflicts with versions of GDAL that may have been installed previously. We will be modifying two environment variables, one called **GDAL_DATA** and the other the system **Path**.

To check for previous GDAL versions:

1. Scroll down in the System variables pane (at the bottom) and look for a variable called **GDAL_DATA**
2. Scroll down further in the System variables pane and find the **Path** variable, click Edit and copy the contents to a text editor, then carefully search through the content and see if there are any references to GDAL
3. Check the User variables pane for the same (less likely that things are here, but it is worth checking)

If **GDAL_DATA** does not exist *and* there are no references to GDAL in the **Path** variable, there should be no conflicts and you can proceed with the below. Otherwise check out the information in step 7 before proceeding.

In the System variables pane, scroll the box down, select the **Path** variable, then go to Edit, hit End to get the cursor at the end, add a semi-colon (;) and then enter the path to your GDAL installation – so, for my installation, I added this:

```
;C:\Program Files (x86)\GDAL
```

Be very careful not to insert any additional spaces before or after the path (see comments)!

Click Ok, then click New and enter the following:

Variable name: **GDAL_DATA**

Variable value is the path to the GDAL data directory – so for me:

```
C:\Program Files (x86)\GDAL\gdal-data
```

To test the installation, open up your command prompt (type cmd in the Run dialogue), type **ogr2ogr** and hit enter. This is one of the programs included with GDAL; if your **Path** is set correctly it will be accessible from anywhere. If some stuff about usage and options is printed out, GDAL is installed

correctly and the Environment variables worked. If it instead says: ‘ogr2ogr’ is not recognized as an internal or external command, operable program or batch file., there is a problem. First try restarting the computer, if ogr2ogr still doesn’t work check your Environment variables for spelling mistakes or typos.

6. Install Python bindings

Run the Python bindings installer downloaded in step 2. If you have multiple Python installations or a non-standard setup, make sure to check which Python the installer is installing to.

To test the entire setup, open up the relevant Python prompt and enter:

```
import ogr
```

Hit enter, then:

```
import gdal
```

If both of these run without printing out anything, you are good to go!

7. Issues/workarounds for existing GDAL installations

If another program has already installed GDAL it might cause problems if you override the Path and GDAL_DATA variables as the library versions may mismatch.

Use information about where the variables point to identify which program installed GDAL, and think carefully about whether you still require it. Ideally you can uninstall the existing version and continue with just one version, if you do this make sure to clean up the Path and GDAL_DATA variables so they point to the correct locations and continue with steps 5 and 6 above. Alternately you might be able to just get the bindings for the version that has already been installed, allowing Python to access that.

If the above are not possible the best solution is to temporarily alter the variables depending on which software is required. Possible ways to do this (none are great for all situations):

- Go through the process of editing your environment variables, as in step 5, each time you need to use the other program (slow, fiddly)
- Create a new Windows user account for one or the other and set the corresponding Path and GDAL_DATA as User variables (not System variables) for the given account – these will take precedence over the System variables and point to the correct locations, but only when that user is logged in (i.e. new user might be PythonGDAL, but there would be errors if a user attempted to use Python with GDAL from a different login)

- The command prompt can temporarily override the system path, which endures as long as that prompt remains open (see below for instructions). This can only be used when Python interpreter/scripts are run from the command prompt.

To temporarily override the system path use the `set` command. If called with no arguments `set` will list the current variables, but its normal usage is `set [variable]=[path]`, for example: `set path=C:\Program Files (x86)\GDAL`. It may be easiest to make this into a batch file which sets both the `Path` and `GDAL_DATA` variables, which is run from the command line before opening the interpreter or running scripts (remember these also will have to be done from the same command line, as the variables are lost when the command prompt is closed). To do this: open Notepad, enter the relevant commands, for example (note that you will probably want to put Python on your path as well, for convenience):

```
1 | set path=C:\Program Files (x86)\GDAL;C:\Python27
2 | set GDAL_DATA=C:\Program Files (x86)\GDAL\gdal-data
```

Save the file as `setPaths.bat` and run this within the command prompt before running scripts.

8. Common issues

When trying “`import gdal`” or “`import ogr`” the `ImportError: DLL load failed: The specified module could not be found`. error is most commonly caused by errors when adding GDALs install folder to the system `Path` variable – go back up to step 5 and triple check everything.

If you have any other problems please read through the comments (below), as they contain some solutions to other issues.

If you are doing Python development, you may be interested in [my Windows Dev Stack](#), which describes my development environment from high level technologies down to specific apps, and how they all work together.

[GDAL](#) [INSTALL](#) [OGR](#) [PATH](#) [PYTHON](#) [WINDOWS](#)

[76](#)

76 thoughts on “OLD METHOD: Installing GDAL / OGR for Python on Windows”

1. Mathias Weibull

says:

[August 29, 2011 at 10:05 am](#)

Hi!

Excellent description, I haven't been able to find anything like your text. Unfortunately something goes wrong when I get to step 5. When trying “`from osgeo import gdal`” I get the following answer:

Python 2.7.1 (r271:86832, Nov 27 2010, 17:19:03) [MSC v.1500 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license" for more information.

```
>>> from osgeo import gdal
```

Traceback (most recent call last):

File "", line 1, in

File "C:\Python27\lib\site-packages\osgeo__init__.py", line 21, in

_gdal = swig_import_helper()

File "C:\Python27\lib\site-packages\osgeo__init__.py", line 17, in swig_import_helper

_mod = imp.load_module('_gdal', fp, pathname, description)

ImportError: DLL load failed: Det gÖr inte att k÷ra %1.

```
>>>
```

best regards,

Mathias

Reply.

1. **StacyR** **AUTHOR**

says:

August 31, 2011 at 9:08 pm

Mathias, I don't know for sure what is causing your error, but if I set my **Path** variable (the one to the GDAL installation folder) incorrectly I get the same error as you... I notice your Python is 64 bit, if you installed 64 bit GDAL you should have added ;C:\Program Files\GDAL to your Path instead of what I had in the tutorial.

If that doesn't work, try posting on [this](#) forum/email thing it helped me out once before...

Reply.

2. **Liz**

says:

September 4, 2013 at 3:01 am

Hey!

So it looks like loads of people have had the same problem. I found a solution on stack exchange but thought it was worth repeating here:

"This was caused because my PATH variable has other paths where a same named dll was used. If you put the c:\program files\gdal before the other paths in PATH, it works fine!"

Thanks for the awesome instructions!

Reply.

1. **Adam**

says:

November 30, 2013 at 10:14 am

The DLL Load Failed... %1 error was caused on our machines by multiple references to a dll in the PATH variable locations. Specifically the C:\Windows\libeay32.dll. PATH order did not matter. Removing the dll causes python load gdal normally. Put it back... error. Solution referenced here:

<http://lists.osgeo.org/pipermail/osgeo4w-dev/2012-June/001781.html>

2. **Luiz Andrade**

says:

September 3, 2014 at 7:23 am

Thank you so much!!!

3. **Christoph**

says:

December 23, 2015 at 2:39 pm

Thanks! Now it's working!

3. **aehostetlerw**

says:

August 17, 2014 at 3:57 pm

After running into the same issue I was able to resolve it by adding the GDAL entry at the beginning of the PATH System Variable instead of at its end.

Thanks for the great instructions. Keep keeping it simple 😊

Reply.

1. **StacyR** **AUTHOR**

says:

September 3, 2014 at 7:52 pm

Thanks for sharing!

2. **Mathias Weibull**

says:

September 12, 2011 at 9:30 am

Hi again. I did that what you recommended but get the same bad things. BUT, when I start python.exe in the C:\Program Files\GDAL it works. I can run the "from osgeo import gdal" but only when starting python from that path. Wrong setup of paths or something?

br,
Mathias

Reply.

1. **StacyR** **AUTHOR**

says:

September 12, 2011 at 1:33 pm

What happens when you go to the windows command line and enter *ogr2ogr*? i.e.:

C:\>*ogr2ogr*

Reply.

3. Mathias Weibull

says:

September 12, 2011 at 7:14 pm

That works fine, and not only from C:\Program Files\GDAL but from other places, too. Which is very confusing... When checking sys.path in python C:\Program Files\GDAL is not included but even when I add it with sys.path.append("C:\Program Files\GDAL") it doesn't work. I wonder if some of the files in C:\Program Files\GDAL should be copied into C:\Python27\lib\site-packages or C:\Python27\lib\site-packages\osgeo ?

Reply.

1. StacyR **AUTHOR**

says:

September 13, 2011 at 9:10 am

For me, sys.path in Python does not contain any of the GDAL paths either – it's all Python related stuff; but 'from osgeo import ogr' does work... Don't copy those files around, it will just make a giant mess of everything!

I strongly suspect the problem is with your GDAL_DATA variable – either: it is incorrect; your installation didn't work properly; or my instructions don't actually work on 64bit systems!

To check system environment variables, list them with 'set' (without quotes, from the windows command prompt). There will be lots of info, but if you scroll to the middle you can confirm that the GDAL_DATA variable exists and is correct.

I just tested out installing 64 bit on my machine (Python 3.2 x64) – it worked fine...

You could also double/triple check you got all the correct installers for your system – the GDAL core and Python bindings were both for the same version of GDAL, and the bindings were for the correct version of Python. Try uninstalling both the bindings and GDAL, download them again and install – making sure to use the 'typical' settings. Delete the two GDAL related changes you made to your environment variables, restart your computer, then add them again (double/triple checking the paths are correct, etc.)...

If none of these things work, I suggest you try a different version of Python or post to the forum I linked to in my first reply (I've run out of ideas!).

Reply.

2. Barry

says:

June 10, 2013 at 5:06 am

I have the same problem as you, it's only work from C:\Program Files\GDAL, at last, I install the GDAL-1.9.1.win-amd64-py2.7 and gdal-19-1600-x64-core, the version is 1.9.1(first I install 1.9.2, but I don't know why the version 1.9.2 can't work), it(1.9.1) work fine.

Reply.

4. Max

says:

October 13, 2011 at 9:02 am

Hi, thanks for your description but I run into the same error message.

```
>>> from osgeo import gdal
```

Traceback (most recent call last):

File "", line 1, in

from osgeo import gdal

File "C:\Python26\ArcGIS10.0\lib\site-packages\osgeo__init__.py", line 21, in

_gdal = swig_import_helper()

File "C:\Python26\ArcGIS10.0\lib\site-packages\osgeo__init__.py", line 17, in

swig_import_helper

_mod = imp.load_module('_gdal', fp, pathname, description)

ImportError: DLL load failed: The specified module could not be found.

But in addition ogr2ogr from command line gives me "'ogr2ogr' is not recognized as an internal or external command operable program or batch file." The strange thing is that "PATH" is definitely set correctly and ogr2ogr runs when I start it from cmd being in the GDAL directory. very strange...

Moved even GDAL into C:\ thinking that "Program Files" may be an issue, but no success...

Max

[Reply](#).

5. **Max**

says:

[October 13, 2011 at 7:14 pm](#)

just found answer to my problem above: No space after or before ";"

max

[Reply](#).

6. **Frankie**

says:

[March 3, 2012 at 5:42 am](#)

Just wanted to say these are fantastic instructions! I was digging around a bit having some issues trying to install based on instructions from other sites when I stumbled upon your site and started from scratch. Everything worked great. Thank you!

[Reply](#).

7. **Annelies**

says:

[May 12, 2012 at 12:30 am](#)

The installation seems easy and I would like to implement it however I cannot access the site

<http://www.gisinternals.com/sdk/>

Does anyone know what happened to the site? Or can someone give me another useful link?

Thanks!!

[Reply](#).

1. **StacyR** **AUTHOR**

says:

May 12, 2012 at 11:24 am

Hey,

looks like the website is down – not sure why, or if/when it might be back! In the mean time you can possibly use the OSGeo4W installer; it allows you to install a whole lot of GIS stuff, including GDAL/OGR. However, I have had a lot of trouble actually using it (program thinks stuff is installed when it isn't, or stuff is said to be successfully installed, but then doesn't work, etc...). Furthermore, the paths and processes you need to follow might be different than what I have said in the above post. I will give it a spin now and see how it works out...

Reply.

1. **Max**

says:

May 13, 2012 at 3:41 am

Hi,

I think you find what you need on this page. I find it especially useful for its 64-bit versions of "everything" <http://www.lfd.uci.edu/~gohlke/pythonlibs/>

max

8. **Annelies**

says:

May 17, 2012 at 1:13 am

Hello,

Thank you for your reply and useful info.

It seems like the site works again, so I'll try to follow the instructions.

Reply.

9. **Bernard**

says:

June 15, 2012 at 7:12 pm

Hi

I did the following:

From <http://www.gisinternals.com/sdk/>

selected MSVC2010 (Win32) -stable > release-1600-gdal-1-9-mapserver-6-0

Downloaded the core and the Py bindings for Py27

Instaled it as per excellant instructions – on a Windows 7 32 bit OS with PYthon2.7

Did typical install for Gdal_core

At the test in Py IDE ("from owgeo import gdal") noticed problem in the __init__.py (for me in C:\Python27\lib\site-packages\osgeo__init__.py)

Then entered some Print statements to init

Result is a pathname as provided by the find_module as follows:

pathname C:\Python27\lib\site-packages\osgeo_gdal.pyd

from “_mod = imp.load_module(‘_gdal’, fp, pathname, description) ” the failure to load the DLL is then reported.

I am new to Python and it seems that the _gdal is to be loaded from the above pathname – or is the _gdal.pyd doing the actual loading?

Also on checking to find a ‘gdal.dll’ only found a “gdal19.dll” and a “gdal16.dll” in the directory where the core gdal installed.

Please assist – is the pathname and file correct for the “pathname” variable?

Thank you very much.

Bernard

Reply.

1. **StacyR** **AUTHOR**

says:

July 26, 2012 at 9:09 am

Hi Bernard,

I think your GDAL_DATA install or Environment Variable might be the problem, which is why it cannot locate the .dll file.

You may have sorted it out already, but if not, try the following:

1. uninstall all the GDAL/OGR stuff and install it again, watch very carefully for erroneous errors, spaces, etc. in your environment variables.
 2. uninstall all the GDAL/OGR stuff, download the latest versions, and install again.
- You really should not interfere with the provided site-packages (i.e. __init__.py)...

Reply.

2. **StacyR** **AUTHOR**

says:

July 26, 2012 at 3:59 pm

Bernard,

do you have PyQt4 installed?

I have had an issue with GDAL and PyQt4 which causes an ImportError to do with a DLL when importing GDAL/OGR. Uninstalling the version of PyQt4 you have installed and replacing it with the version here: <http://www.lfd.uci.edu/~gohlke/pythonlibs/#pyqt> seems to work for me, tested and verified for Python 2.6 (32bit) and Python 3.2 (64bit).

Reply.

10. usablebeauty

says:

August 21, 2012 at 3:04 am

That worked perfectly for me, thanks!

Reply

1. Bernard

says:

September 20, 2012 at 2:52 am

Thanks Staccy – I reinstalled the package several times and experimented to form simpler short directories so as to avoid errors in the environment var. Also downloaded the packages numerous times. No luck.

Nate that Google Sketchup also installed GDAL (earlier version.) I also removed this and reinstalled with no luck. I did not check to see if any problem in registry.

So changed to C++. If I have time I shall experiment with a way to get Python to read the library via own interface and then step it aoo the way to see what is going wrong. Thanks. Bernard

Reply

1. Simon

says:

October 10, 2012 at 2:00 am

I experienced the same problem. When running python from the folder containing the gdal binaries gdal19.dll and a lot of other dll's which gdal19 was linked against, I could import gdal, but not from any other location. So I guessed it had to do with the order in which things appeared in my PATH variable. For example I had PyQt4 installed which also contained a libmysql.dll, and the PyQt4-stuff appeared before the gdal-folder in my PATH. So if I changed my PATH so that the gdal-folder appeared first, everything works like it should.

I dont think it has to do with other gdal environment variables (like GDAL_DATA which is only used by certain drivers and to translate certain reference systems..)

11. Irini

says:

October 16, 2012 at 11:23 pm

Thank you very much... your advises saved me for sure!!! I was completely confused... you was sooooo helpful!!! Thanks again!!!

Reply

12. Bernard

says:

January 11, 2013 at 6:01 am

Hi

After a good dose of C++ and qt4 had some time during my vacation to revisit this Python~GDAL issue. Carefull investigated the process – then proceded as follows:

Uninstalled all Python and GDAL packages. Reinstalled Py3.2 (the exact same installation package previously used.) Continued with GDAL core. Then added the reference to the GDAL to system path variable (see cmd prompt display below.) Then started the Py3.2 IDE and displayed the "system" path;

```
>>> import sys
>>> print (sys.path)
['C:\\Python32\\Lib\\idlelib', 'C:\\Windows\\system32\\python32.zip',
'C:\\Python32\\DLLs', 'C:\\Python32\\lib', 'C:\\Python32', 'C:\\Python32\\lib\\site-packages']
>>>
```

Then checked the system path variable from Windows command prompt:

```
Microsoft Windows [Version 6.1.7600] Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Genman01>path PATH=C:\Program Files\GDAL\;C:\Program Files\PC Connectivity
Solution\;...;c:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\PrivateAssemblies\
C:\Users\Genman01>path^A >> import sys
>>> print (sys.path)
['C:\\Python32\\Lib\\idlelib', 'C:\\Program Files\\GDAL',
'C:\\Windows\\system32\\python32.zip', 'C:\\Python32\\DLLs', 'C:\\Python32\\lib',
'C:\\Python32', 'C:\\Python32\\lib\\site-packages']
>>>
```

Now installed the python bindings for the particular gdal core.

Then started the Py3.2 IDE and at last:

```
>>> from osgeo import gdal
>>>
```

In general, I noticed that Py can also obtain path information from *.pth (hope I remember this correct) – I was unable to obtain any clear documentation on the use of this files and exactly how info is taken and added to the existing Py sys.path.

I hope this may be of use to others as well.

PS: Please note, I'm not very good in programming, actually doing this as I am looking at putting some GIS in place for a foundation I am setting up with aim to further scientific knowledge and care for environment where I live. If any person interested I could sure do with help. Aim to map distribution map of species of all plants and animals in my area. Also would like to get a model going for invasive species (mainly plants and ants.) one of other important projects is getting to plant a million trees under the "Plant for the planet" program. Should in next couple of months get a website going for the "Outeniqua Biodiversity Foundation"

Bernard

Reply.

13. **Bernard**
says:

January 11, 2013 at 6:26 am

Hi

Sorry for the above, it was mangled due to the “capital 6” special character.

After a good dose of C++ and qt4 had some time during my vacation to revisit this Python~GDAL issue. Carefull investigated the process – then proceded as follows:

Uninstalled all Python and GDAL packages. Reinstalled Py3.2 (the exact same installation package previously used.) Continued with GDAL core. Then added the reference to the GDAL to system path variable (see cmd prompt display below.) Then started the Py3.2 IDE and displayed the “system” path;

```
>>> import sys
>>> print (sys.path)
['C:\\Python32\\Lib\\idlelib', 'C:\\Windows\\system32\\python32.zip',
'C:\\Python32\\DLLs', 'C:\\Python32\\lib', 'C:\\Python32', 'C:\\Python32\\lib\\site-
packages']
>>>
```

Then checked the system path variable from Windows command prompt:

```
Microsoft Windows [Version 6.1.7600] Copyright (c) 2009 Microsoft Corporation. All rights reserved.
C:\Users\Genman01>path PATH=C:\Program Files\GDAL\;C:\Program Files\PC Connectivity
Solution\;...;c:\Program Files\Microsoft Visual Studio 9.0\Common7\IDE\PrivateAssemblies\ C:
... >> import sys
>>> print (sys.path)
['C:\\Python32\\Lib\\idlelib', 'C:\\Program Files\\GDAL',
'C:\\Windows\\system32\\python32.zip', 'C:\\Python32\\DLLs', 'C:\\Python32\\lib',
'C:\\Python32', 'C:\\Python32\\lib\\site-packages']
>>>
```

Now installed the python bindings for the particular gdal core.

Then started the Py3.2 IDE and at last:

```
>>> from osgeo import gdal
>>>
```

In general, I noticed that Py can also obtain path information from *.pth (hope I remember this correct) – I was unable to obtain any clear documentation on the use of this files and exactly how info is taken and added to the existing Py sys.path.

I hope this may be of use to others as well.

PS: Please note, I’m not very good in programming, actually doing this as I am looking at putting some GIS in place for a foudation I am setting up with aim to further scientific knowledge and care for environment were I live. If any person interested I could sure do with help. Aim to map distribution map of species of all plants and animals in my area. Also would like to get a mocel going

for invasive species (mainly plants and ants.) one of other important projects is getting to plant a milion trees 9under the “Plant for the planet” program. Should in next couple of months get a website going for the “Outeniqua Biodiversity Foundation”

Bernard

Reply.

14. **Dragec**

says:

February 26, 2013 at 12:08 am

Hi.

Your how-to works fine, but I need ogr2ogr with OCI driver.

ogr2ogr on <http://www.gisinternals.com/sdk/> doesn't have OCI driver compiled.
Is there version with OCI enabled?

I also tried to install gdal with OCI from osgeo4w, but this didn't work for me on Windows_7 x64.
I just can't run the ogrinfo.exe – Application was unable to start correctly.

Reply.

15. **maria**

says:

May 14, 2013 at 11:40 pm

Dear Colleges,

I am trying to install GDAL for the Python application. I need GDAL since I want to convert some points to raster grid and I have read, it is possible by using GDAL.

The Python version is 2.7. I have followed your instruction and when I say ogr2ogr it works but when I say 'from osgeo import gdal'

I receive the error: DLL load failed. I would be thankful if you kindly help me with it.

I have checked the environment and it is correct. what would be the problem in your idea?

Reply.

1. **StacyR** **AUTHOR**

says:

May 21, 2013 at 1:12 pm

I don't think you correctly added the GDAL_DATA environment variable (either you made a mistake with the path, or didn't name it correctly)...

Cheers,
Stacy

Reply.

16. Pingback: [PythonでGeoSpatialをやってみる。環境設定編\[Chapter 1\] | OpenなGISのこと](#)

17. **James**

says:

July 5, 2013 at 3:31 pm

Hi,

I get the following error message when entering from osgeo import ogr

following the above instructions:

Traceback (most recent call last):

File "", line 1, in

File "C:\Python27\ArcGIS10.1\lib\site-packages\ogr.py", line 2, in

from osgeo.gdal import deprecation_warn

File "C:\Python27\ArcGIS10.1\lib\site-packages\osgeo__init__.py", line 21, in

_gdal = swig_import_helper()

File "C:\Python27\ArcGIS10.1\lib\site-packages\osgeo__init__.py", line 17, in

swig_import_helper

_mod = imp.load_module('_gdal', fp, pathname, description)

ImportError: DLL load failed: The specified procedure could not be found.

Cheers

James

Reply.

18. **Jenna Mack**

says:

July 7, 2013 at 8:00 am

Thanks for these instructions...overall they are great!

I'm running into one problem that I know is specific to my computer but I'm not sure how to resolve...any advice would be greatly appreciated.

Everything works fine until Step 5 when I try to install the bindings. I receive an error saying that Python 2.6 must be installed and that it cannot be found in the registry.

I actually do have Python 2.6.5 installed as part of ArcGIS in the following folder
C:\Python26\ArcGIS10.0

With this version of Python, in this location I have already successfully installed NumPy and SciPy and they both found this version in the registry.

Is there a work around so that this GDAL install will find the same registry information?

Thanks!

Jenna

Reply.

1. **StacyR** **AUTHOR**

says:

July 8, 2013 at 2:02 pm

Try this:

from the GIS Internals page download the GDAL and Mapserver source (i.e. release-XXX-gdal-X-XX-mapserver-X-X-src.zip) matching the GDAL library you have already installed. Unzip it and navigate into the folder, then you need to open up a command window in this GDAL folder. In Windows XP you can do this by right clicking on the folder, while in newer versions of Windows you need to hold down shift while you right click. Now check you can access your Python install by entering the full path to python.exe, i.e., "C:\Python26\ArcGIS10.0\python.exe" (no quotes), then hitting enter. This ought to open up the interpreter. Type "exit()", followed by enter, to get back to the command prompt. Now type "C:\Python26\ArcGIS10.0\python.exe setup.py install" and hit enter. This ought to install the bindings to the Python that you called to open the setup file...

Let me know how you get on!

Reply

19. Pingback: GDAL – Maptember +3 | geosmiles

20. **Micha**

says:

September 21, 2013 at 8:18 am

I also followed the great instructions on installing GDAL. All seemed to work fine. After setting the path and running the test on the command prompt all seemed fine. However, when running from the python interpreter it did not work – reporting that the DLL could not be loaded.

I then changed the path variable, putting the GDAL path as the first entry (be careful to avoid extra spaces and put the semi-colon in the right place). It now works.

An extra tip. When testing from a command prompt or other python interpreter be sure to restart the prompt or interpreter program (I use Pythonxy and Spyder) – programs read the environment on starting. Hope this helps avoid the frustration I had!

Reply

1. **StacyR** **AUTHOR**

says:

September 22, 2013 at 9:29 am

GDAL requires some other libraries; I think this error can occur when another program has added a different version of a library to the path. Moving GDAL to the start will mean that GDAL's library version comes up first, but may cause problems when running the other program!

I have had this issue with PyQt4.

Reply

21. **Robin St.Clair (@Uberseehandel)**

says:

October 25, 2013 at 2:40 am

Thanks so much for this. I followed your instructions exactly and it worked fine. After a day and a half of struggling, I can now feel the stress leaking away.

Fantastic!

Reply.

22. Pingback: [How to run OGR in ArcGIS Scripts? | Question and Answer](#)

23. **Max**

says:

February 14, 2014 at 2:02 am

Thanks for all that advice — I installed exactly the same setup on a second computer, but also had the problem of “ImportError: DLL load failed” and from command prompt “ImportError: Module not found” Moving the path variable in front did not help either.

It worked after I finally reinstalled Python 2.7 — not sure where this DLL conflict comes from, but maybe a corrupt one in my Python installation? In any case, if you have the same issue, also try reinstalling Python itself

Reply.

24. **Valentas**

says:

May 4, 2014 at 12:57 am

I am not able to load <http://www.gisinternals.com/sdk/> this week! Is there any other place in the world you can download GDAL core files and python bindings? Please do not advice <http://www.lfd.uci.edu/~gohlke/pythonlibs/> as the version of GDAL available there does not work on 32 bit windows.

Reply.

1. **StacyR** **AUTHOR**

says:

September 3, 2014 at 7:45 pm

Unfortunately it seems to go down every now and again – seems to come back in a few days...

Reply.

25. **John**

says:

June 10, 2014 at 8:41 pm

Runtime error

Traceback (most recent call last):

File “”, line 1, in

ImportError: No module named osgeo

Any advice as to what i have done wrong mate?

Reply.

1. **StacyR** **AUTHOR**

says:

September 3, 2014 at 7:46 pm

Python cannot access the bindings – have you installed them to the actual Python you are trying to use?

Reply.

26. **abner245@gmail.com**

says:

June 13, 2014 at 2:22 am

THANK YOU, THANK YOU, THANK YOU for posting this. Amazing!

Reply.

1. **Catch**

says:

November 25, 2014 at 2:49 am

I also append my thanks! After a whole day of trying to install this, I've found your site and than about half an hour later it's ready to go! THANK YOU 😊

Reply.

27. **Rares**

says:

August 10, 2014 at 2:59 am

Thank you for the clear steps and examples, it works.

I had a problem with the PATH (error DLL from importing the module while running a .py script, containing import gdal). I solved it by changing the order in the PATH. (GDAL path placed in front of others).

Reply.

1. **StacyR** **AUTHOR**

says:

September 3, 2014 at 7:51 pm

Thanks for sharing!

Reply.

2. **StacyR** **AUTHOR**

says:

September 13, 2015 at 11:58 pm

I have added some more detail to the post to help out with this – it may not always be the best idea to change the path order, if there are other important things going on.

Reply.

28. Pingback: Conflict between ArcPy and GDAL/OGR? | Question and Answer

29. **Gin**

says:

August 26, 2014 at 10:56 pm

I had the same problem as comment #1 and I solved using this thread in Stack Overflow:

<http://stackoverflow.com/questions/6009506/unable-to-install-python-and-gdal-dll-load-failed>

Basically, it is VERY IMPORTANT, to match the compiler version of Python, the Python binding and the GDAL Core package. In my case, I downloaded version 1600 of Python binding and GDAL core package because they were the latest at this moment (08/2014). However, the compiling version of my Python was 1500.

Additionally, I confused architectures: My laptop is a 64 bits machine, HOWEVER, my Python version is 32 bits. So I had to find Python binding packages + GDAL core packages for Win 32 bits architecture AND compiling version 1500. Both things.

Hope this helps! 😊

Reply.

1. **StacyR** **AUTHOR**

says:

September 13, 2015 at 11:57 pm

Thanks Gin, I have never had an issue with different compiler versions (at the moment my Python is v.1500 and GDAL 1800, working fine). However, getting the correct architecture is critical, and I have added some more information about how to determine this to the post.

Reply.

30. **Kenn**

says:

September 25, 2014 at 4:13 am

Beautiful how-to. I followed through with installation and it worked like magic. Thanks very much.

Reply.

31. **MrZogs**

says:

October 1, 2014 at 7:03 am

I have the same error and i tried all stuff that it is mention above but nothing work. I install another realease in my case since i have win8 64bits i install release-1600-x64-gdal-1-11-0-mapserver-6-4-1 which correspond (according to the present date) to te last row of "GDAL and MapServer latest release versions". It's a dummy solution i know.. but it was a workaround.

Reply.

32. Pingback: [How To Fix Windows 8 Error Loading Python Dll in Windows](#)

33. Pingback: [How To Fix Python Error Window in Windows](#)

34. **Jamie**

says:

February 12, 2015 at 5:53 am

Like many others who posted here, I was getting errors in Python saying the DLL failed to load, and/or the module could not be found, when trying to import gdal. The solution for me, which was also mentioned above by Gin on Aug.26, 2014 was to install the correct Python compiling version. Like Gin, I have a 64-bit computer, but I am running 32-bit version of Python, so I installed 32-bit 1600 compiled version of GDAL with appropriate Python bindings. This combination did not work, so I tried the 32-bit 1500 compiled version of GDAL with appropriate Python bindings and now it works!! YAY!!

Thanks to Gin for her posting, it helped me and probably others, and thanks to this web page and the GISInternals Support Site for supplying software and know-how for users in need.

[Reply](#).

1. **StacyR** **AUTHOR**

says:

September 13, 2015 at 11:55 pm

Thanks Jamie,

I have added a bit more background info to the post.

[Reply](#).

35. **Michel**

says:

April 2, 2015 at 10:19 am

I had problems installing the bindings after a colleague found out they were installed on d: drive for some reason. That is why I was getting the traceback import error. It worked after copying the files from D:\Lib\site-packages to C:\Python32\Lib\site-packages.

[Reply](#).

36. **striker3945**

says:

June 16, 2015 at 9:16 pm

Your instructions worked fine, however when I try to open a .he5 file with gdal.Open, I get the following error:

RuntimeError: ... not recognised as a supported file format

[Reply](#).

1. **StacyR** **AUTHOR**

says:

September 13, 2015 at 11:54 pm

Search for the supported GDAL file formats – this appears unrelated to the installation, which is what this post describes.

[Reply](#).

37. **David**

says:

June 18, 2015 at 5:19 am

I thought I'd share my experience. Great instructions by the way! I'm on a Windows 7 64-bit machine, but with 32-bit Python 2.7, so I installed the 32-bit versions of the files in this How To. I was trying all the tests (i.e. 'ogr2ogr' and 'from osgeo import ogr') in the Windows command prompt. The ogr2ogr worked, from the basic, non-python directory. Then I started Python in CMD (I just typed 'python' pressed enter, and it becomes a Python interactive window) and tried the 'from osgeo import ogr', it said "No module named osgeo". Bummer. Then after reading some on the internet, I decided to open up my go to Python editor program, Komodo Edit 9.1. This has an auto-complete/suggestion function and when I typed 'import' it then gave a list of all the modules I could import, which

included gdal. So I typed 'import gdal', clicked run, and nothing happened! This meant it worked. Just to be sure I double checked with the following code and again it appeared to work, as indicated by the text 'imported GDAL'.

```
try:
import gdal
print 'imported GDAL'
except:
print 'NO GDAL!!'
```

This is as far as I've gotten, but I think I'm good to go now!

Reply.

1. **StacyR** **AUTHOR**

says:

September 13, 2015 at 11:53 pm

David,

You must have multiple Python installations. One Python is on your system Path (i.e. in your Path environment variable), which is the one that comes up when you enter 'python' at the command prompt. However, your bindings are installed to another Python install, which is the one used by Komodo.

Reply.

38. **Alexey**

says:

August 18, 2015 at 4:49 am

Hello to all! Thanks for instructions. My attempts to install on WIN7 64bit finally were successful with next combination (have no idea why) on different computers:

python-2.7.8.amd64.msi

gdal-1.11.1800-x64-core.msi

GDAL-1.11.2.win-amd64-py2.7.msi

with all environment variable changes as at top of this site.

Thank again.

Alexey.

Reply.

39. Pingback: [Enable Python installations that were not installed by ArcGIS to access Arcpy functionality | Python, GIS and stuff...](#)

40. **Hafiz**

says:

September 17, 2015 at 10:44 pm

I had same problem it is simply solved when i change the file name "_gdal.pyd" to _gdal.py"

Reply.

41. **Cindy**

says:

December 23, 2015 at 9:57 am

Anyone have try to install GDAL on windows 10? anu specific problem to it? Thanks!

Reply.

42. **Hynek**

says:

January 4, 2016 at 7:19 am

I prefer 64-bit GDAL, but have to use ArcPy as well, so I have installed 32-bit GDAL as well. This created a problem, because changing values of PATH and GDAL_DATA variables didn't work, and I couldn't import OGR and GDAL while working in PyScripter.

So I created this short script, that adjusts the values and allows me to import 64-bit GDAL, while the default setting is set for 32-bit GDAL. It's quite simple and I think it's better, than to manually rewrite values of variables, when you want to use 64-bit GDAL.

Simply copy the content into your script, adjust paths (I'm using Windows 7), and it should work.

See <http://pastebin.com/Xi8qfdyp>

Reply.

43. Pingback: Utöka GIS med Python – Del 3 | Geosupportsystem

44. **Shruti**

says:

April 7, 2016 at 8:28 am

Hi, I want GDAL/OGR in to Python which is installed by ArcGIS; how can I install it?

Reply.

1. **StacyR** **AUTHOR**

says:

April 13, 2016 at 8:01 am

Hi there, try the new method of installing GDAL/OGR for Python here:

<https://pythongisandstuff.wordpress.com/2016/04/13/installing-gdal-ogr-for-python-on-windows/>

Regards,

Stacy

Reply.

45. Pingback: Installing GDAL / OGR for Python on Windows | Python, GIS and stuff...

46. **Jonas**

says:

September 6, 2017 at 3:34 am

May I ask a stupid question?

When I start python it displays:

Python 2.7.13 | packaged by conda-forge | (default, May 2 2017, 13:28:48) [MSC v.1500 64 bit (AMD64)] on win32

Am I correct to disregard the "win32" at the end and only consider the 64?

(I.e. I clicked on "release-1500-x64-gdal-2-2-1-mapserver-7-0-6" on the website, and then on "gdal-202-1500-x64-core.msi" and "GDAL-2.2.1.win-amd64-py2.7.msi".)

Reply.

1. **StacyR** **AUTHOR**

says:

September 6, 2017 at 5:53 am

You are correct to ignore the “win32”, your Python install is 64 bit

Reply.

[Blog at WordPress.com.](#)