

Calling arcpy from an external virtual Python environment

Over time, one's python environment can grow to include a number of non-standard modules. While great for doing work locally, this can become difficult to keep track of when developing code that you give to others ("which modules do I need to ship with my software?"). In these types of situations, it can be handy to build a standalone virtual Python environment that calls on your platform's version of arcpy. It's pretty easy to set this up. The steps are: 1) make sure you have the python virtualenv.exe executable on your machine, 2) use it to create a virtual environment directory, and 3) augment that so it knows about your platform's ArcGIS content.



Note that using the Python third-party tool virtualenv.exe is one of several approaches. This "virtual environment" setup can be replicated using other products, such as the Anaconda "conda" tool. The biggest key is to ensure that your (non-ESRI) python environment is similar enough to Esri's shipped environment that arcpy will import and run properly.

The first two steps are well covered [in this web posting](#). An important addition to these instructions for this use case is to use the ESRI version of python that came with your platform's ArcGIS if you find that virtualenv.exe is not already on your machine. virtualenv.exe, if present, is located in the Scripts subdirectory of your ESRI python home. The ESRI python home can be determined by starting ArcMap and typing "import sys; sys.prefix". If virtualenv.exe is missing, you'll use the pip.exe program to install it. If pip.exe itself is missing from the Scripts subdirectory, that will need to be installed (it never ends, does it?). To install pip.exe, you will need to download and run a python script from the web posting. When you run that script, use the ESRI python.exe (start ArcMap and type "import sys; sys.prefix" to find it).

One thing that might be confusing is that the virtualenv.exe and pip.exe programs are run from the MS-DOS prompt and not from within Python. The gist of virtualenv.exe is that it creates a directory based on the name you specify, actually copying things like the python.exe and associated materials into subdirectories below what you specified. This way your virtual environment will be based on your "vanilla" python environment. You can then use pip.exe to subsequently augment it with any non-standard modules you need for the job at hand. These will be stored in the Lib\site-packages directory. Prior to starting python, you then "activate" this virtual environment at the MS-DOS prompt.

To augment this virtual environment for use with arcpy, you don't want to try to copy all the ESRI stuff into your new directory. You just want to tell it where to find the ArcGIS installed components. To do this, you'll need to place a *.pth file w/in the Lib\site-packages subdirectory. We'll call it "zzArcGIS.pth". In the file, you'll need to put all the directories that ArcGIS uses to make arcpy work. The easiest way to get the not-small list of pathnames is to start ArcMap, type "import sys; sys.path", and copy the list into zzArcGIS.pth. Note that the file should look something like the following.

```
# zzArcGIS.pth
# Path to ArcGIS arcpy modules
# Place in folder ...\
```

Then, after using ...\