Readme for Geologic Names Check tool

The Geologic Names Check Tool provides a cursory check of the formal geologic nomenclature applied to the names of geologic units listed in a GeMS-compliant DescriptionOfMapUnits table (DMU). It allows for the semi-automated comparison of the usage, age, and extent (to the State level) of the unit names with the information compiled in Geolex.

This tool is a Python script intended to be executed along with the GeMS Validate Database script. The results of these two scripts provide the information necessary for map authors and managers of the U.S. National Geologic Map Database to determine whether a geologic map database is in compliance with the GeMS schema (see https://ngmdb.usgs.gov/Info/standards/GeMS/).

The script first queries the DMU for the map unit names and related content. The script then queries the U.S. Geologic Names Lexicon (Geolex, https://ngmdb.usgs.gov/Geolex/) to identify possible matches with any formal geologic names that are among the map unit names in the GeMS file. Finally, the script creates a spreadsheet in Excel format for review by the map author. After inspection and author comments have been completed, the spreadsheet is to be submitted to USGS along with the results of the GeMS validation script. The secretary of the Geologic Names Committee (GNC) then will evaluate and compile into the U.S. geologic names lexicon (Geolex) information from the author-provided comments and references and(or) from the current map (when published).

Installation

The Geologic Names Check tool is part of the GeMS Tools toolbox of ArcGIS geoprocessing tools. There are two versions depending on software; one for use with ArcGIS Pro and one for ArcMap. Development of the toolboxes is managed through Github. To install, first go to the appropriate Github repository:

ArcGIS Pro: https://github.com/usgs/gems-tools-pro

ArcMap: https://github.com/usgs/gems-tools-arcmap

The screenshots below all reference the ArcGIS Pro version but the process is the same for the ArcMap version.

Click on the green *Code* button and choose *Download ZIP*.
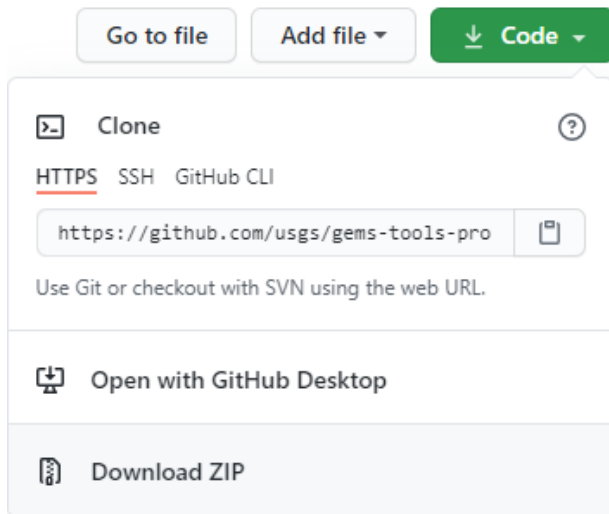
Choose 'Download ZIP' unless you are familiar with using Git to manage repositories.

Even if you have no interest in collaborating on the code, Git can still be useful if you just want to quickly get the latest version of the toolbox. Use

`git clone` https://github.com/usgs/gems-tools-pro.git --depth 1

once to set up the repository. This performs a shallow clone that does not include the full commit history. Get updates with
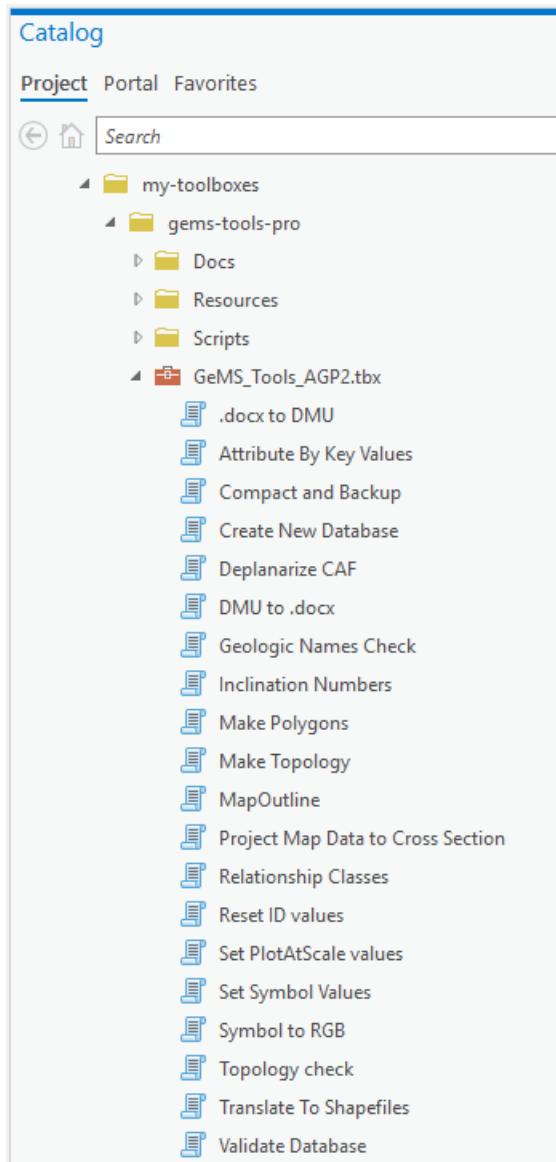
`git pull --depth 1`

Save the zip file to a location that makes sense for your workflow and unzip it. It will extract to a folder called gems-tools-pro-master (or gems-tools-arcmap-master), within which will be the folders and files required for the tools.
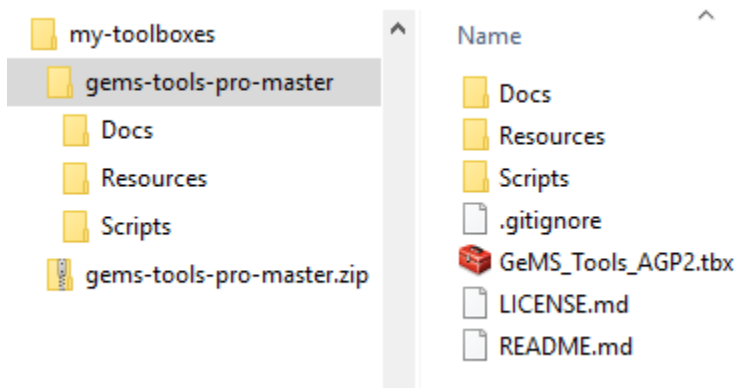
The organization of the toolbox contents as seen in Windows Explorer.

Note that .gitignore is only useful if you will be using Git to maintain your copy of the files. You may delete it if you wish with no effect on the tools

In ArcGIS Pro or ArcMap, go to the Catalog pane, navigate to the folder containing the .tbx file, expand the contents of the toolbox, and you will see the tools.

Catalog

Project Portal Favorites

← 🏠 | Search

▲ 📁 my-toolboxes
  ▲ 📁 gems-tools-pro
    ▷ 📁 Docs
    ▷ 📁 Resources
    ▷ 📁 Scripts
    ▲ 🧰 GeMS_Tools_AGP2.tbx
        📄 .docx to DMU
        📄 Attribute By Key Values
        📄 Compact and Backup
        📄 Create New Database
        📄 Deplanarize CAF
        📄 DMU to .docx
        📄 Geologic Names Check
        📄 Inclination Numbers
        📄 Make Polygons
        📄 Make Topology
        📄 MapOutline
        📄 Project Map Data to Cross Section
        📄 Relationship Classes
        📄 Reset ID values
        📄 Set PlotAtScale values
        📄 Set Symbol Values
        📄 Symbol to RGB
        📄 Topology check
        📄 Translate To Shapefiles
        📄 Validate Database

View of the toolbox contents in ArcGIS Pro as of 10/27/20

The contents of the ArcMap toolbox will be slightly different as a result of the different underlying versions of the ArcPy Python packages available to each program.
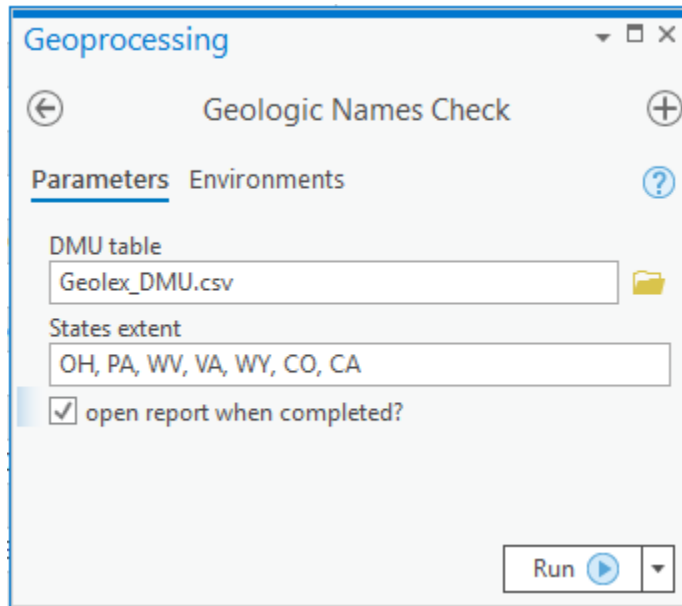
Parameter Form Usage

The tool takes three arguments:

DMU table – A GeMS-compliant DescriptionOfMapUnits table. It may be a table in a file geodatabase, an Excel spreadsheet, or text file (comma or tab delimited). The table must contain the fields HierarchyKey, MapUnit, Fullname, and Age, but they may be in any order, need not be CamelCase, and there are no spceial constraints on how the values are formatted. Values in Name are searched for Geolex names. If no Geolex name is found, Fullname is searched as well.

States extent – One or more comma delimited state abbreviations that define the region over which the DMU is relevant.

Open report when completed? – True or false to open the Excel report file once the script has completed. This value is True by default and optional.

The Geologic Names Check tool parameter form in ArcGIS Pro.

Geolex_DMU.csv ships with the toolbox and can be found in the Resources folder. It is provided to show examples of the results to expect from the tool depending on different map unit names and does not represent a realistic stratigraphy.

The value here of 'States extent' is for demonstration purposes only. When using Geolex_DMU.csv you may change the abbreviations as you like to see how the results change. The full set of abbreviations that cover all usages in Geolex_DMU.csv is:
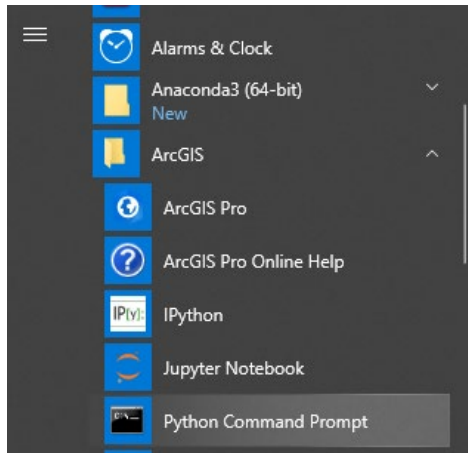
AK, AL, AR, AZ, CA, CO, CT, ID, IL, IN, KY, MD, MI, MO, MS, NJ, NM, NV, NY, OH, OK, OR, PA, TN, UT, VA, WA, WI, WV, WY

When run, the parameter form calls GeMS_GeolexCheck_AGP2.py in the Scripts folder. ArcGIS Pro uses Anaconda to manage Python 3 environments and packages. The default installation should include all the site packages required to run the tool. Messages reporting the progress of the tool are printed to the geoprocessing results window.

The arguments are processed slightly differently in ArcMap. The original Python script was written in Python 3.7.9 against site packages that are part of the default installation with ArcGIS Pro 2.5 but are not installed with the version of Python 2.7 that ships with ArcMap. To ease the burden of configuration for ArcMap users, all necessary packages were bundled into a single executable file (GeMS_GeolexCheck.exe in the Resources folder) which takes the same arguments as the original script. The ArcMap parameter form is tied to a Python script (GeMS_GeolexCheck_Arc10.py in the Scripts folder) that simply collects the arguments and sends them to GeMS_GeolexCheck.exe as if it were being called at the command line. This will prompt a Windows command prompt window to open within which most of the progress messages are displayed. When the processing has finished, press any key to return to ArcMap.

Command Line Usage – ArcGIS Pro

The simplest way to run GeMS_GeolexCheck_AGP2.py in an environment that has the right dependencies installed is to open the Python Command Prompt that is configured during the installation of ArcGIS Pro (use the Windows search bar or browse to it from the Windows Start icon to the left of the search bar)

This will open a window with a prompt that shows the active Conda environment (by default this is arcgispro-py3) followed by the current working directory:

```
(arcgispro-py3) C:\ArcGIS\Pro\bin\Python\envs\arcgispro-py3>
```

The command needs to take this form:

```
python <path to GeMS_GeolexCheck_AGP2.py> <path to DMU> <states extent> <True | False>
```

As with the parameter form, the last argument is optional.

For example:

```
(arcgispro-py3) C:\ArcGIS\Pro\bin\Python\envs\arcgispro-py3>python C:\_AAA\my-toolboxes\gems-tools-pro\Scripts\GeMS_GeolexCheck_AGP2.py C:\_AAA\my-toolboxes\gems-tools-pro\Resources\Geolex_DMU.csv "OH, PA, WV, VA, WY, CA, CO"
```

Any argument that contains spaces, whether a path or the states extent string, must be enclosed within double quotes.

Tables inside file geodatabases or Excel spreadsheets must specified as a path term after the path to the .gdb or .xlsx file:

```
C:\projects\WallaceQuad\Wallace.gdb\DescriptionOfMapUnits
```

“`C:\projects\GromitQuad\GromitDMU.xlxs\Description of MapUnits`” (note use of double quotes)

Command Line Usage – ArcMap

The steps to be taken to run the Python script for use with an ArcMap installation are largely the same as those for use with ArcGIS Pro with one caveat. You must either

1) make sure that python.exe is in your PATH environment variable, or
2) at the command prompt, change the directory to the location of python.exe

Editing the PATH environment variable is beyond the scope of this readme, but if you choose to go this route keep two things in mind:

1) if you don't have admin rights to your computer, you can still edit environment variables for your account, although you may need to first create a PATH variable
2) if you have installed 64 bit background geoprocessing installed (see https://desktop.arcgis.com/en/arcmap/10.7/analyze/executing-tools/64bit-background.htm) edit your PATH variable to point to C:\Python27\ArcGISx6410.7, otherwise use C:\Python27\ArcGIS10.7

The Python script to call is GeMS_GeolexCheck_Arc10.py in the Scripts folder. Remember, the purpose of this script is to collect and verify the arguments supplied in the ArcMap parameter form in order to send them to GeMS_GeolexCheck.exe in the Resources folder.

Command Line Usage – GeMS_GeolexCheck.exe

The bundled executable file GeMS_GeolexCheck.exe may be called from the command line or by any scripting language that has access to system commands. This offers a completely open-source method for evaluating DMUs that does not require the installation of any ESRI software or Python. The EXE file can still evaluate tables in file geodatabases provided that the location of GDAL binaries are in the PATH environment variable, otherwise only Excel and comma or tab delimited text files are usable. Because the EXE file was created on a Windows 10 computer (using Pyinstaller), the file will only run on Windows 10.

The EXE takes the same arguments as the Python scripts, for example:

```
C:\projects\my-toolboxes\gems-tools-arcmap-master\Resources>
GeMS_GeolexCheck.exe Geolex_DMU.csv "OH, PA, WV, VA, WY, CA, CO"
```

Results

The result of running the tool is a report in Excel format with a name that ends in "_namescheck.xlsx". If the input DMU is in a file geodatabase, the report file is named the same as the geodatabase; otherwise, the name is the name of the Excel spreadsheet or CSV.

See GeologicNamesCheck_report_README.pdf for detailed explanation of the sections of the report and how to perform the review.

Build environments

	ArcMap	ArcGIS Pro
Operating system	Windows 10	Windows 10
ArcGIS	10.7.1	2.5.0
Python	2.7.14	3.6.9

An internet connection is required to query the Geolex api.