

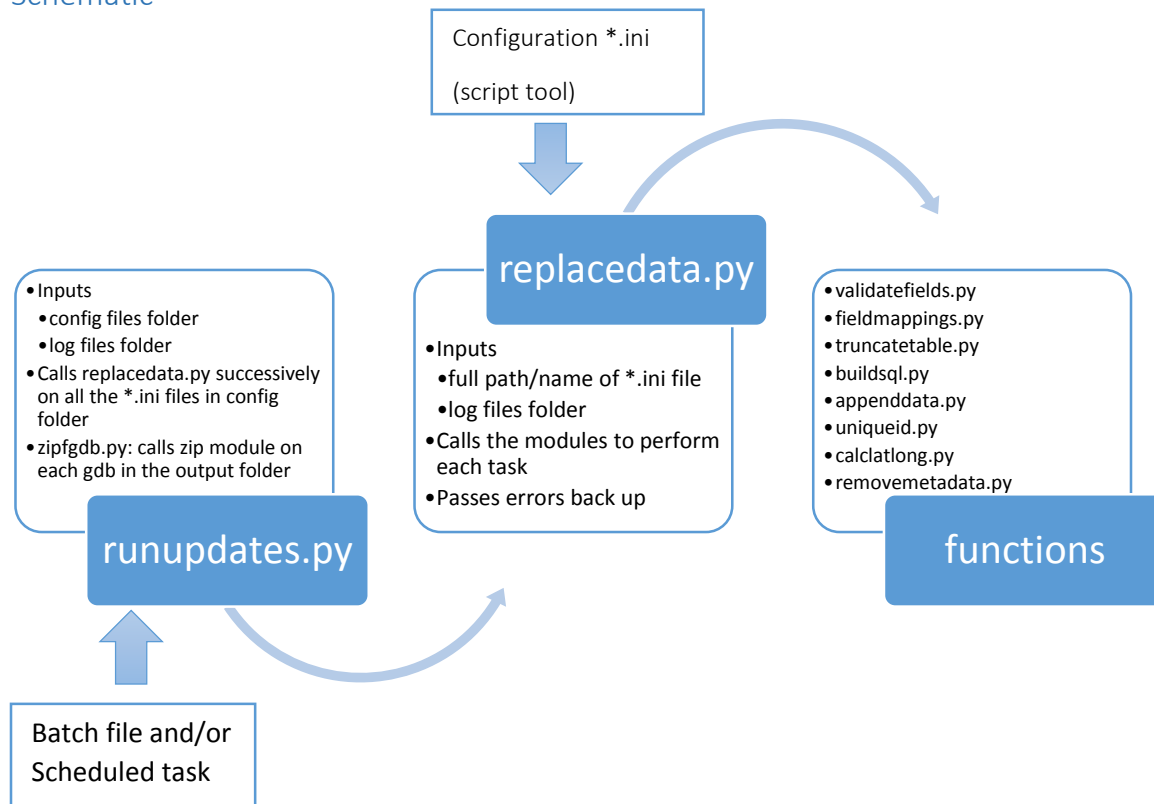
Data Preparation for Integration Automation

Updated April 10, 2016

The following document describes the setup for the data integration automation data preparation. The output will be a zipped file geodatabase (containing the data integration layers) that is ready to be emailed or dropped into Google Drive. There is a folder in our shared Drive called “Consolidated Geodatabases”.

It is up to each agency whether they want to schedule the process to run in the background on a regular basis, or run it manually as needed.

Schematic



Summary

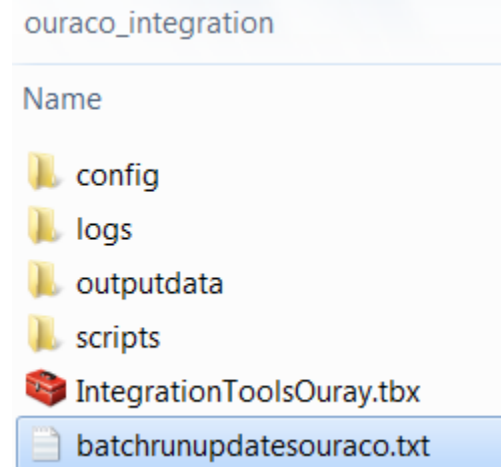
- Step 1: Set up environment
- Step 2: Configure using script tools
- Step 3: Create batch file to run script
- Step 4 (optional): Set up a task to run updates on a schedule
- Step 5: Monitoring success/failure

Requirements

Computer to execute scripts/batch file which has Python 2.x and ArcGIS 10.1+ (for arcpy library)

Step 1: Environment

Unpack the zip file into an easily findable and permanent location. It contains the following folder structure, which you can place under a root folder such as `ouraco_integration`.



These locations can all be local or on your server, but must be accessible to the computer which will execute the script (i.e. has Python & ArcGIS installed). Best not to have any spaces in the paths if you can avoid it.

***If you move/rename your source data**, you will need to run the configuration script tools again with the new locations. If you move the integration scripts folder structure, you will also need to update the batch file paths (see step 3) and possibly your scheduled task with the path to the batch file (see step 4). It would be optimal to find a good home for the folders before running the configurations.

The `config` folder is where all your config files (*.ini files) will be located.

The `logs` folder holds the log files.

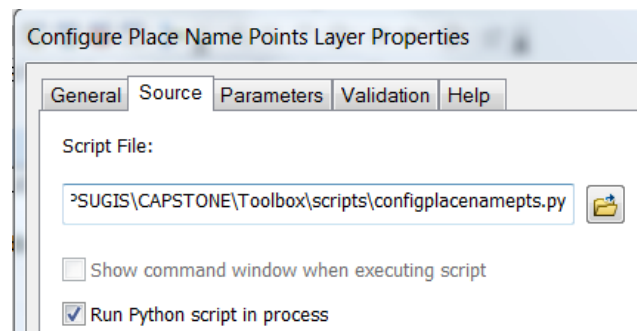
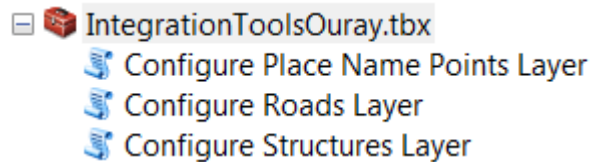
`Outputdata` is the location of the “blueprint” geodatabase I am providing.

`Scripts` is the folder for the scripts.

The toolbox holds the script tools used for configuration.

Step 2: Configuration

Navigate to the location of the Integration Toolbox using ArcCatalog. **Go to the properties** of each script tool and **set the script source** (right click/Properties/Source tab) to where the scripts are located. It may be correct already, but check.



`configaddresspts.py` – for access points and structure points: tool can be copy/pasted (in ArcCatalog) and then set with different data source defaults for the two layers, which share a schema. Or just run it again with a different name for the *.ini file and different data sources/outputs.
`configplacenamepts.py` – for place name points
`configroads.py` – for roads
`configesns.py` – for ESNs, if applicable

Optional: You may also choose to set up script tool defaults in the parameters tab, such as your agency code/SOD.

Run each script tool in the Integration Toolbox to set up a different configuration file (*.ini file) for each layer. Each layer has a different script tool because of the different field mappings. Right now there are tools for access points, structure points (same tool), place name points and roads.

Use the tool help to see what the tool is looking for in each box.

In the top box, you will need to **paste in the path to the config folder and give the .ini file a name**, such as `places.ini`. For example, `c:\mypath\ouraco_integration\config\places.ini`.

If you have additional data querying ("Additional SQL") to do besides `SOD="YOURAGENCYCODE"` (already coded in), I can help you with the formatting for this box, such as: `AND COUNTY="SAN MIGUEL"`

For the field mapping section (below datum transformation), if there isn't a field that matches the schema in your data, delete the text in the corresponding box. If your field has a different name, use the pull-down menu to select it. Below are screenshots of running the script tools.

Place Names layer configuration - blank

Configure Place Name Points Layer

Configuration File

Input Feature Class

Output File Geodatabase

Output Feature Class

Agency Code

Additional SQL (optional)

Datum Transformation
NAD_1983_To_WGS_1984_5

Source of Data (optional)
SOD

State Id (optional)
STATE_ID

Site Id (optional)
SID

Date Last Updated (optional)
DLU

Name (optional)
NAME

Alias (optional)
ALIAS

Alias 2 (optional)
ALIAS2

Site Address Number (optional)
SAN

Site Address Number Suffix (optional)
SNS

Street Prefix Direction (optional)
PRD

Configure Place Name Points Layer

Create and write configuration file for data sources, SQL expression and field mapping for place name points layer for West Region GIS data integration.

OK Cancel Environments... << Hide Help Tool Help

Place Name points configuration after choosing Input Feature Class - note warning

Configure Place Name Points Layer

Click error and warning icons for more information

Configuration File
C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\config_files\ouray\places.ini

Input Feature Class
C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\sent\METSA_OURACO_PLACE_NAM

Output File Geodatabase
C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\output\OURACO.gdb

Output Feature Class
C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\output\OURACO.gdb\PLACES

Agency Code
OURACO

Additional SQL (optional)

Datum Transformation
NAD_1983_To_WGS_1984_5

Source of Data (optional)
SOD

State Id (optional)
STATE_ID

Site Id (optional)
SID

Date Last Updated (optional)
DLU

Name (optional)
NAME

Alias (optional)
ALIAS

Alias 2 (optional)
ALIAS2

Site Address Number (optional)
SAN

Site Address Number Suffix (optional)
SNS

Street Prefix Direction (optional)

Configuration File

Path and file name of *.ini file that will be written to by the script. If the file exists, it will be overwritten. For example:
C:\Temp\placenames.ini

OK Cancel Environments... << Hide Help Tool Help

Place Name configuration errors - These two fields have different names in the input than the default. Simply choose the correct field name from the dropdown. **If there isn't a matching field, delete the text in the box.**

SOURCE_QC

Collection Date (optional)
COLLECTIONDATE

Upload Date (optional)
UPLOADDATE

Quality (optional)
QUALITY

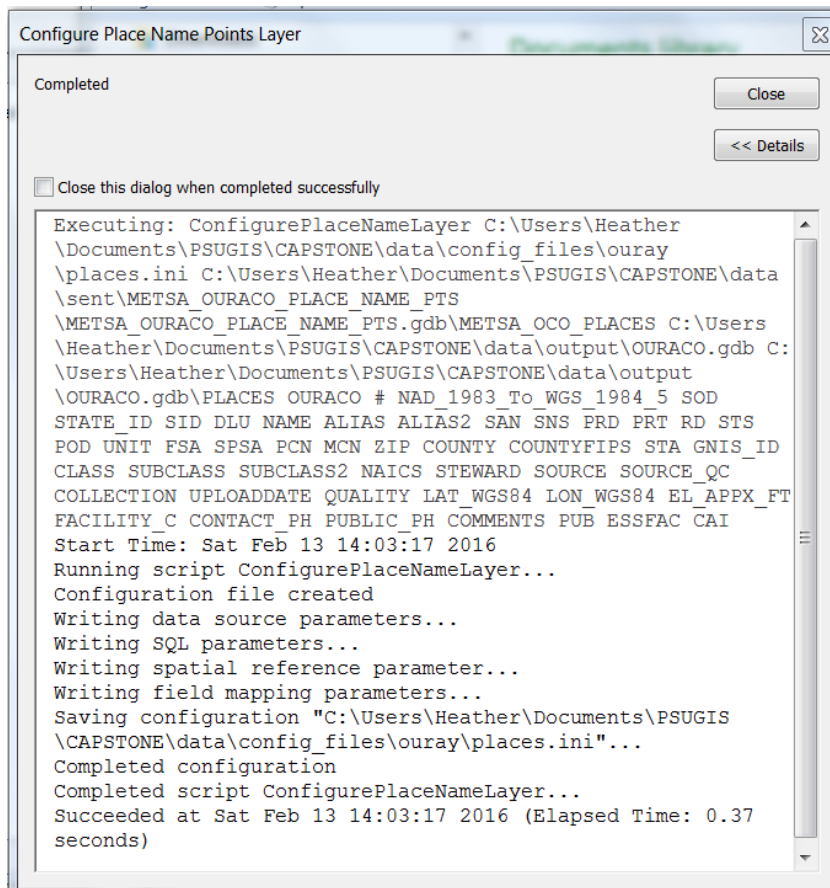
Latitude (optional)
LAT_WGS84

Longitude (optional)
LON_WGS84

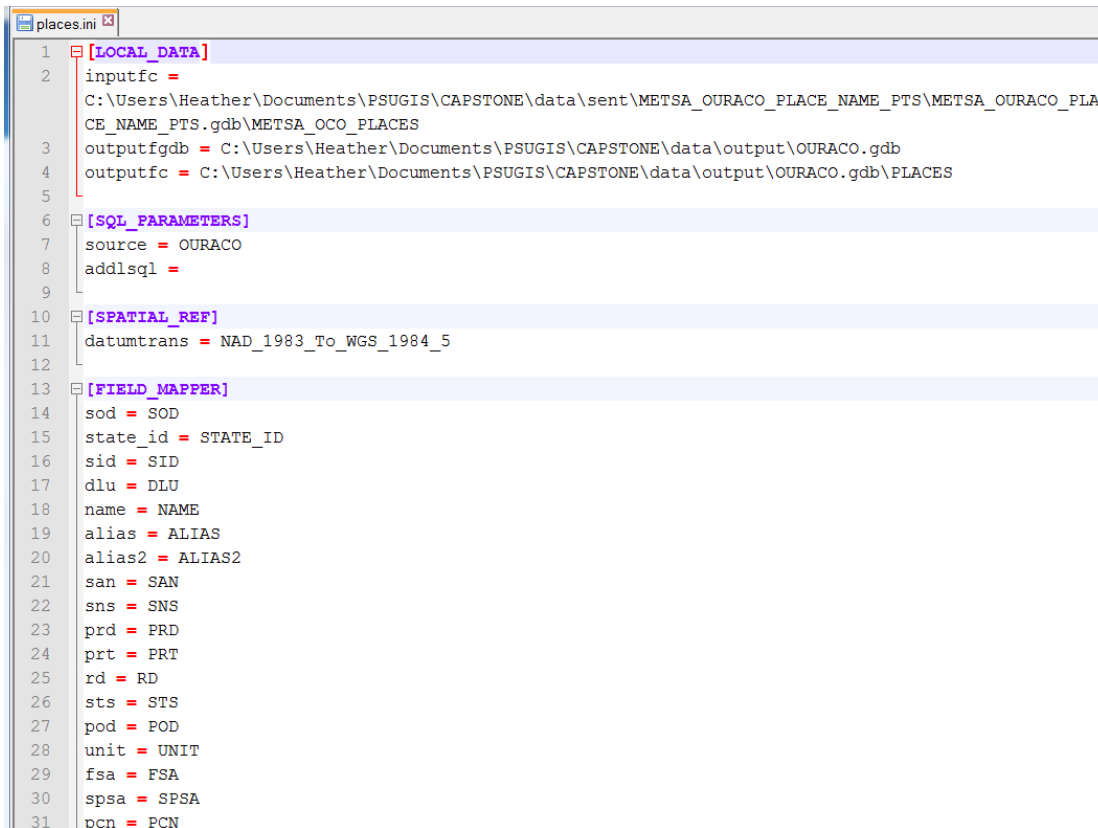
Elevation (approx. ft.) (optional)
EL_APPX_FT

Facility Contact (optional)
FACILITY_CONTACT

Takes a fraction of a second to complete



Place Names configuration (.ini) file written by the config script tool. Can be viewed in a text editor. If you need to make changes, simply run the config script tool again OR you can perform edits in the text editor if you're careful.



```
1 [LOCAL_DATA]
2 inputfc =
3 C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\sent\METSA_OURACO_PLACE_NAME_PTS\METSA_OURACO_PLA
4 CE_NAME_PTS.gdb\METSA_OCO_PLACES
5 outputfgdb = C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\output\OURACO.gdb
6 outputfc = C:\Users\Heather\Documents\PSUGIS\CAPSTONE\data\output\OURACO.gdb\PLACES
7
8 [SQL_PARAMETERS]
9 source = OURACO
10 addlsql =
11
12 [SPATIAL_REF]
13 datumtrans = NAD_1983_To_WGS_1984_5
14
15 [FIELD_MAPPER]
16 sod = SOD
17 state_id = STATE_ID
18 sid = SID
19 dlu = DLU
20 name = NAME
21 alias = ALIAS
22 alias2 = ALIAS2
23 san = SAN
24 sns = SNS
25 prd = PRD
26 prt = PRT
27 rd = RD
28 sts = STS
29 pod = POD
30 unit = UNIT
31 fsa = FSA
32 spsa = SPSA
33 pcn = PCN
```

Step 3: Batch file

I have included a batch file template. **Change the extension of the batch file to “.bat” instead of “.txt”** and **open in a text editor** such as NotePad or NotePad++ (not MS Word).

This is the only hard-coded part of these scripts. The file requires:

1. Path to your Python installation (depends on your ArcGIS version)
2. Path of the `runupdates.py` script
3. Path to config files folder
4. Path to logs folder
5. Path to data output folder

See example below. There is a single space between each parameter, not a carriage return. Note that the command is all on one line - line 4 in the screenshot below. If there are spaces in any of the paths, enclose each whole path in double quotes: `"c:\my\path has\spaces"`. The script and folders can be located anywhere as long as the computer running the batch file has Python and ArcGIS (for the `arcpy` library) installed.

I have written simple error checking in the batch file which will leave the black command window open if there is an error, so you know to check the logs.

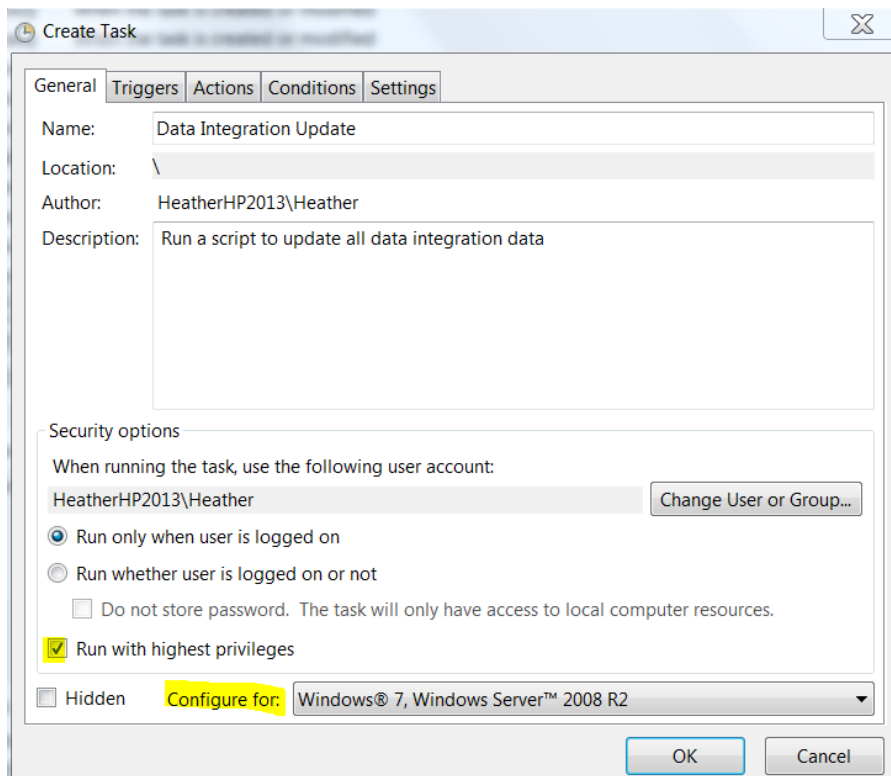
If you don't want to set up a scheduled task, you can double click the .bat file (in Windows File Explorer) to run it manually on a computer with Python & ArcGIS installed.

```
1 @ ECHO OFF
2 REM run integrated gis data update script
3 echo Running data integration script runupdate.py...
4 C:\Python27\ArcGIS10.3\Python.exe
   C:\Users\Heather\Documents\PSUGIS\CAPSTONE\Toolbox\ouraco_integration\scripts\runupdate.py
   C:\Users\Heather\Documents\PSUGIS\CAPSTONE\Toolbox\ouraco_integration\config
   C:\Users\Heather\Documents\PSUGIS\CAPSTONE\Toolbox\ouraco_integration\logs
   C:\Users\Heather\Documents\PSUGIS\CAPSTONE\Toolbox\ouraco_integration\outputdata
5 if %errorlevel% neq 0 (
6     echo Data integration script error %errorlevel%: check logs
7     pause
8 )
9 exit
```

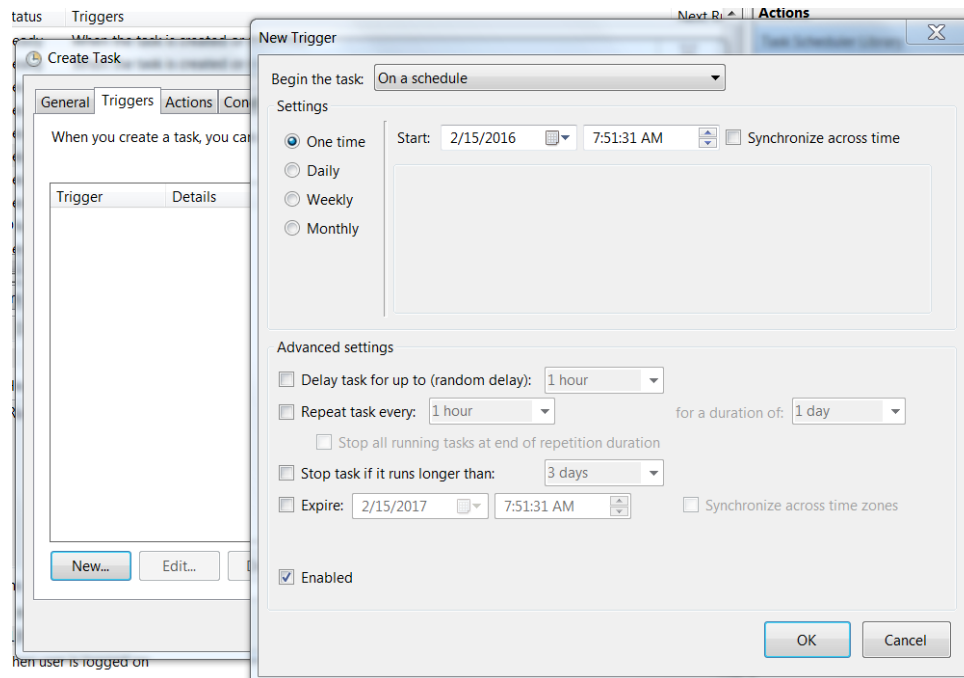
Step 4: Scheduled task

At the Windows start menu search box, **type Task Scheduler, right click and Run as Administrator.**

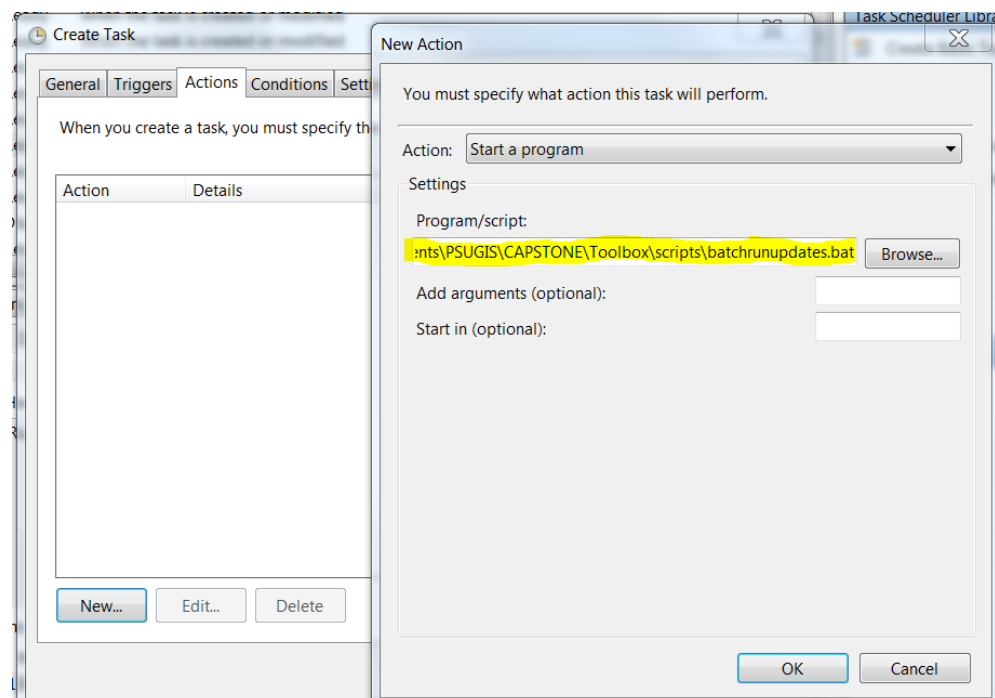
Create a task (not a basic task), choosing when to run, how often and what to run (your bat file).



Go to the Triggers tab and set up a schedule if desired. You can also *not* create a trigger, to run manually.



On the Actions tab, set the action to Start a Program, and then browse or paste in the path to your batch file. No arguments are necessary.





You can experiment with the other settings, but those are the ones that matter. The task may pop up a command window during execution, which you can ignore. If the window stays open, there has been an error.

Step 5: Monitoring success/failure

I have configured the batch file to stay open with a warning message if the main script fails. At that point, you would go into the logs folder and review the logs. Also, I have created a function to send you a message via email if you have the ability to send SMTP email on your network. We would edit the sendmessage.py file with your SMTP server hostname, port, and email address.

There are two log files, which are created and updated automatically. The warn log holds only warnings and errors. The debug log has all the details. These files can be viewed in a text editor.

 debuglog.log

 warnlog.log

The following are samples from the two logs. They have the timestamp, the level of the logging message, the name of the module generating the entry, and the message. The warn log will be empty if there are no warnings or errors.

```
debuglog.log - Notepad
File Edit Format View Help
2016-04-06 15:38:51,687 - INFO - runupdate - *-----
2016-04-06 15:38:51,687 - INFO - runupdate - Beginning updater script to update all data.
2016-04-06 15:38:51,687 - DEBUG - runupdate - *-----
2016-04-06 15:38:51,691 - DEBUG - replacedata - Reading configuration file for... C:\User
\PSUGIS\CAPSTONE\Toolbox\ouraco_integration\config\places.ini
2016-04-06 15:38:54,940 - DEBUG - replacedata - Checking field lengths
2016-04-06 15:38:55,651 - DEBUG - replacedata - Succeeded: validate fields
2016-04-06 15:38:55,651 - DEBUG - replacedata - Field mapping
2016-04-06 15:38:56,512 - DEBUG - replacedata - Succeeded: field mapping
2016-04-06 15:38:56,512 - DEBUG - replacedata - Truncating table
2016-04-06 15:38:58,466 - DEBUG - replacedata - Succeeded: truncate table
2016-04-06 15:38:58,566 - DEBUG - replacedata - Building SQL expression
2016-04-06 15:38:58,908 - DEBUG - replacedata - PLACES | Final SQL expression: SOD = 'OUR
2016-04-06 15:38:58,914 - DEBUG - replacedata - Appending data
2016-04-06 15:39:00,296 - DEBUG - replacedata - Succeeded: append data
2016-04-06 15:39:00,296 - DEBUG - replacedata - Checking for unique IDs
2016-04-06 15:39:00,427 - DEBUG - replacedata - Checked for unique IDs
2016-04-06 15:39:00,427 - DEBUG - replacedata - Repairing geometry
2016-04-06 15:39:00,834 - DEBUG - replacedata - Repaired geometry
2016-04-06 15:39:01,026 - DEBUG - replacedata - Calculating lat and long fields
2016-04-06 15:39:02,526 - DEBUG - replacedata - Succeeded: calculate lat and long fields
2016-04-06 15:39:02,526 - DEBUG - replacedata - Removing unwanted metadata and gp history
2016-04-06 15:39:03,901 - DEBUG - replacedata - Succeeded: removed unwanted metadata and
2016-04-06 15:39:03,901 - DEBUG - runupdate - Succeeded: replace data for places.ini
2016-04-06 15:39:03,901 - DEBUG - runupdate - *-----
2016-04-06 15:39:03,907 - DEBUG - replacedata - Reading configuration file for... C:\User
\PSUGIS\CAPSTONE\Toolbox\ouraco_integration\config\roads.ini
2016-04-06 15:39:04,309 - DEBUG - replacedata - Checking field lengths

warnlog.log
1 2016-04-08 13:22:45,005 - WARNING - replacedata - ROADS | Invalid or null ids found (13)
2 Duplicate ids found (5)
3 GUNNCO_3516
4 GUNNCO_2341
5 GUNNCO_788
6 GUNNCO_788
7 GUNNCO_788
```