

Running and Scheduling QGIS Processing Jobs

QGIS Tutorials and Tips



Author

Ujaval Gandhi

<http://www.spatialthoughts.com>

Running and Scheduling QGIS Processing Jobs

You can automate a lot of tasks in QGIS using Python scripting (PyQGIS) and the Processing Framework. Most of the time, you would run these scripts manually while QGIS is open. While that is helpful, many times you need a way to run this jobs via the command-line and without needing to open QGIS. Fortunately, you can write standalone python scripts that use QGIS libraries and can be run via the command-line. In this tutorial, we will learn how to write and schedule a job that uses the QGIS Processing framework.

Overview of the task

Let's say we are working on some analysis using shapefiles of a region. The shapefiles are updated on a daily basis and we always need the latest file. But before we can use these files, we need to cleanup the data. We can setup a QGIS job that automates this process and runs it daily so you have the latest cleaned up shapefiles for your work. We will write a standalone Python script that downloads a shapefile and run topological cleaning operations on a daily basis.

Other skills you will learn

- Downloading and unzipping files using Python.
- Running any Processing algorithm via PyQGIS.
- Fixing topological errors in a vector layer.

Get the data

[Geofabrik](#) provides daily updated shapefiles of [OpenStreetMap](#) datasets.

We will use [shapefiles for Fiji](#) for this exercise. Download the [fiji-latest.shp.zip](#) and unzip it to a folder on your disk.

Data Source [GEOFABRIK]

Procedure

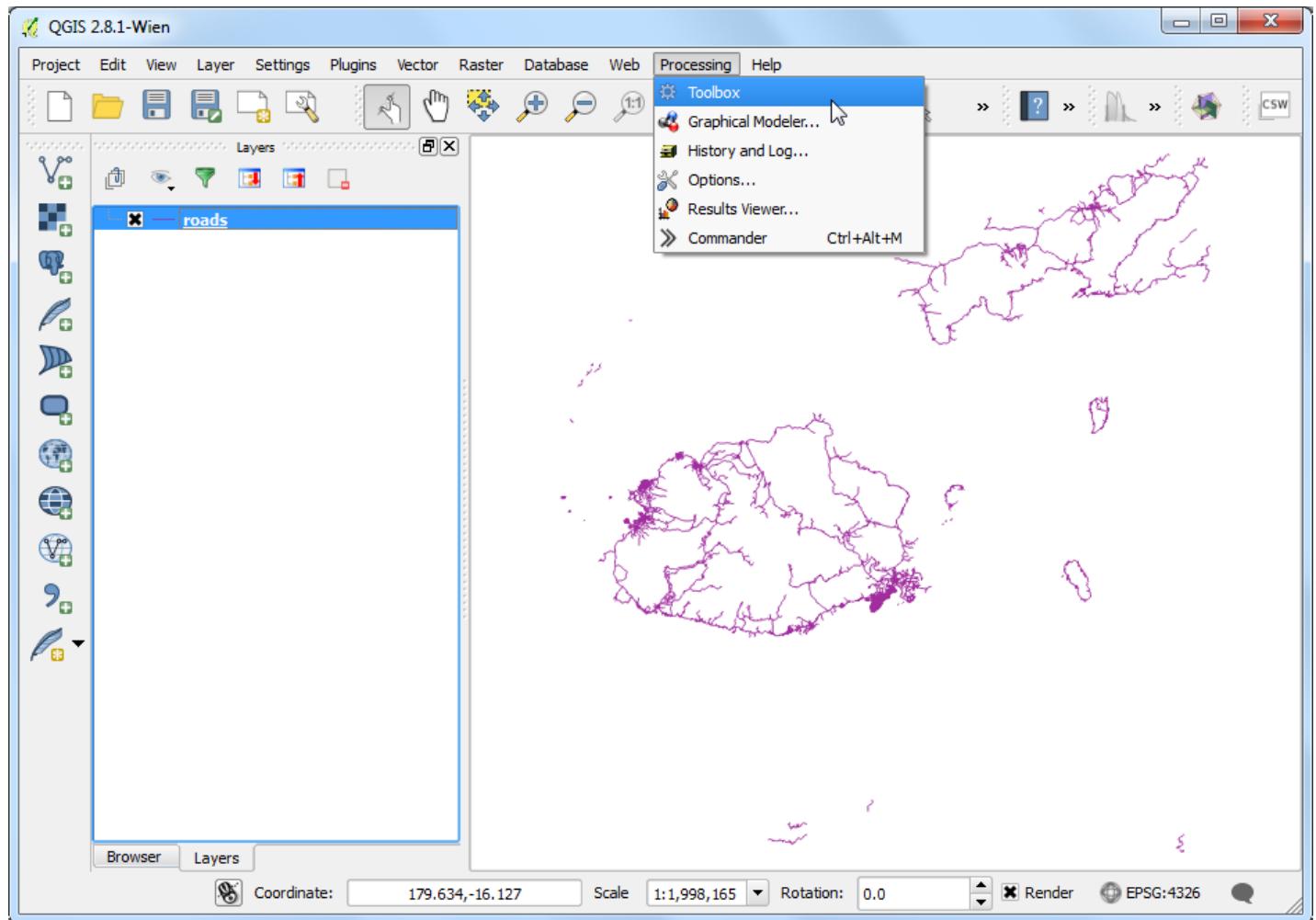
1. We will first run through the process of cleaning the shapefile manually to note the commands that we will use in the python script. Launch QGIS and go to Layer ▶ Add Layer ▶ Add Vector Layer.



2. Browse to the folder containing the unzipped shapefiles and select the `roads.shp` file and click Open.



3. First we must re-project the roads layer to a Projected CRS. This will allow us to use *meters* as units when performing analysis instead of degrees. Open Processing > Toolbox.



4. Search for the Reproject layer tool. Double-click it to launch the dialog.



5. In the Reproject layer dialog, select the `roads` layer as Input layer. We will use EPSG:3460 Fiji 1986 / Fiji Map Grid CRS as the Target CRS. Click Run.



- Once the process finishes, you will see the reprojected layer loaded in QGIS. Go to Processing > History and Log...



7. In the History and Log dialog, expand the Algorithm folder and select the latest entry. You will see the full processing command shown in the bottom panel. Note this command for use in our script.



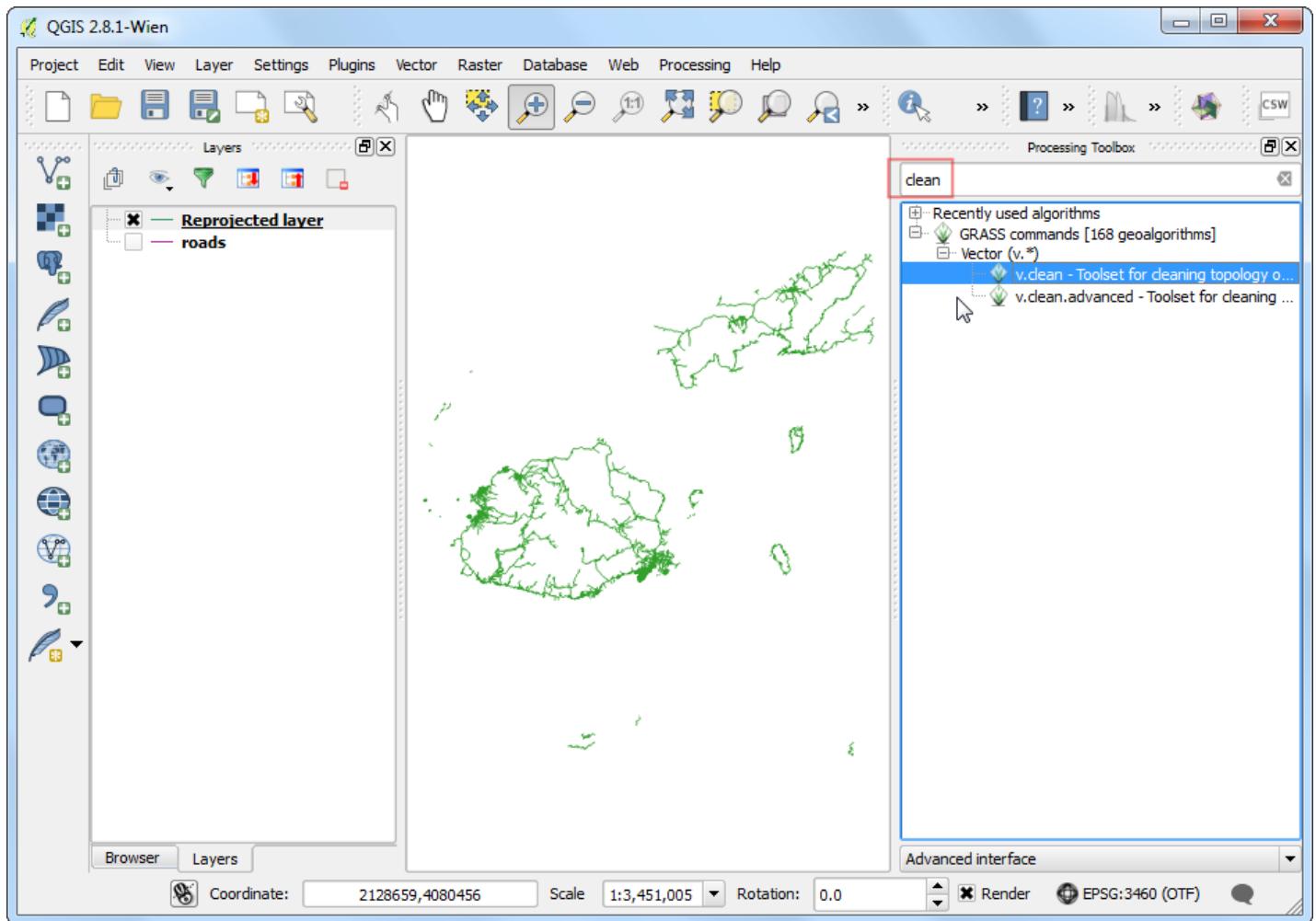
8. Back in the main QGIS Window, click at the CRS button in the bottom-right corner.



9. In the Project Properties | CRS dialog, check the Enable on-the-fly CRS transformation and select EPSG:3460 Fiji 1986 / Fiji Map Grid as the CRS. This will ensure that our original and reprojected layers will line up correctly.



10. Now we will run the cleaning operation. GRASS has a very powerful set of topological cleaning tools. These are available in QGIS via the v.clean algorithm. Search for this algorithm in the Processing Toolbox and double-click it to launch the dialog.



11. You can read more about various tools and options in the Help tab. For this tutorial, we will be using the snap tool to remove duplicate vertices that are within 1 meter of each other. Select Reprojected layer as the Layer to clean. Choose snap as the Cleaning tool. Enter 1.00 as the Threshold. Leave the other fields blank and click Run.



12. Once the processing finishes, you will see 2 new layers added to QGIS. The Cleaned vector layer is the layer with topological errors corrected. You will also have a Errors layer which will highlight the features which were repaired. You can use the errors layer as a guide and zoom in to see vertices that were removed.



13. Go to Processing > History and Log dialog and note the full processing command for later use.



14. We are ready to start coding now. See the **A Text Editor or a Python IDE** section in the [Building a Python Plugin](#) tutorial for instructions to setup your text editor or IDE. For running standalone python scripts that use QGIS, we must set various configuration options. A good way to run standalone scripts is to launch them via a .bat file. This file will first set the correct configuration options and then call the python script. Create a new file named `launch.bat` and enter the following text. Change the values according to your QGIS configuration. Don't forget to replace the username with your own username in the path to the python script. The paths in this file will be the same on your system if you installed QGIS via the OSGeo4W Installer. Save the file on your Desktop.

Note

Linux and Mac users will need to create a shell script to set the paths and environment variables.

```
REM Change OSGEO4W_ROOT to point to the base install folder
SET OSGEO4W_ROOT=C:\OSGeo4W64
SET QGISNAME=qgis
SET QGIS=%OSGEO4W_ROOT%\apps\%QGISNAME%
set QGIS_PREFIX_PATH=%QGIS%
REM Gdal Setup
set GDAL_DATA=%OSGEO4W_ROOT%\share\gdal\
```

```

REM Python Setup
set PATH=%OSGEO4W_ROOT%\bin;%QGIS%\bin;%PATH%
SET PYTHONHOME=%OSGEO4W_ROOT%\apps\Python27
set PYTHONPATH=%QGIS%\python;%PYTHONPATH%

REM Launch python job
python c:\Users\Ujaval\Desktop\download_and_clean.py
pause

```

```

C:\Users\Ujaval\Desktop\launch.bat - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
launch.bat x
1 REM Change OSGEO4W_ROOT to point to the base install folder
2 SET OSGEO4W_ROOT=C:\OSGeo4W64
3 SET QGISNAME=qgis
4 SET QGIS=%OSGEO4W_ROOT%\apps\%QGISNAME%
5 set QGIS_PREFIX_PATH=%QGIS%
6 REM Gdal Setup
7 set GDAL_DATA=%OSGEO4W_ROOT%\share\gdal\
8 REM Python Setup
9 set PATH=%OSGEO4W_ROOT%\bin;%QGIS%\bin;%PATH%
10 SET PYTHONHOME=%OSGEO4W_ROOT%\apps\Python27
11 set PYTHONPATH=%QGIS%\python;%PYTHONPATH%
12
13 REM Launch python job
14 python c:\Users\Ujaval\Desktop\download_and_clean.py
15 pause

```

Batch file length : 477 lines : 15 Ln : 15 Col : 6 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS

15. Create a new python file and enter the following code. Name the file as download_and_clean.py and save it on your Desktop.

```

from qgis.core import *
print 'Hello QGIS!'

```

```
C:\Users\Ujava\Desktop\download_and_clean.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
launch.bat download_and_clean.py
1 from qgis.core import *
2
3 print 'Hello QGIS!'

Python file length : 46 lines : 3 Ln : 3 Col : 20 Sel : 0 | 0 Dos\Windows UTF-8 w/o BOM INS .
```

16. Switch to your Desktop and locate the launch.bat icon. Double-click it to launch a new command window and run the script. If you see Hello QGIS! printed in the command window, your configuration and setup worked fine. If you see errors or do not see the text, check your launch.bat file and make sure all the paths match the locations on your system.



17. Back in your text editor, modify the `download_and_clean.py` script to add the following code. This is the bootstrap code to initialize QGIS. These are unnecessary if you are running the script within QGIS. But since we are running it outside QGIS, we need to add these at the beginning. Make sure you replace the username with your username. After making these changes, save the file and run `launch.bat` again. If you see `Hello QGIS!` printed, you are all set to do add the processing logic to the script.

```

import sys
from qgis.core import *

# Initialize QGIS Application
QgsApplication.setPrefixPath("C:\\\\OSGeo4W64\\\\apps\\\\qgis", True)
app = QgsApplication([], True)
QgsApplication.initQgis()

# Add the path to Processing framework
sys.path.append('c:\\\\Users\\\\Ujaval\\\\.qgis2\\\\python\\\\plugins')

# Import and initialize Processing framework
from processing.core.Processing import Processing
Processing.initialize()
import processing

print 'Hello QGIS!'

```

The screenshot shows a Notepad++ window with two tabs: 'launch.bat' and 'download_and_clean.py'. The 'download_and_clean.py' tab contains the following Python code:

```
1 import sys
2 from qgis.core import *
3
4 # Initialize QGIS Application
5 QgsApplication.setPrefixPath("C:\\OSGeo4W64\\apps\\qgis", True)
6 app = QgsApplication([], True)
7 QgsApplication.initQgis()
8
9 # Add the path to Processing framework
10 sys.path.append('c:\\Users\\Ujaval\\.qgis2\\python\\plugins')
11
12 # Import and initialize Processing framework
13 from processing.core.Processing import Processing
14 Processing.initialize()
15 import processing
16
17 print 'Hello QGIS!'
```

The status bar at the bottom of the Notepad++ window displays: length:465 lines:17 Ln:10 Col:62 Sel:0|0 Dos\Windows UTF-8 w/o BOM INS.

18. Recall the first processing command that we had saved from the log. This was the command to re-project a layer. Paste the command to your script and add the surrounding code as follows. Note that processing commands return the path to the output layers as a dictionary. We are storing this as the `ret` value and printing the path to the reprojected layer.

```
roads_shp_path = "C:\\\\Users\\\\Ujaval\\\\Downloads\\\\fiji-latest.shp\\\\roads.shp"
ret = processing.runalg('qgis:reprojectlayer', roads_shp_path, 'EPSG:3460',
None)
output = ret['OUTPUT']
print output
```

```
1 import sys
2 from qgis.core import *
3
4 # Initialize QGIS Application
5 QgsApplication.setPrefixPath("C:\\OSGeo4W64\\apps\\qgis", True)
6 app = QgsApplication([], True)
7 QgsApplication.initQgis()
8
9 # Add the path to Processing framework
10 sys.path.append('c:\\Users\\Ujaval\\qgis2\\python\\plugins')
11
12 # Import and initialize Processing framework
13 from processing.core.Processing import Processing
14 Processing.initialize()
15 import processing
16
17 roads_shp_path = "C:\\Users\\Ujaval\\Downloads\\fiji-latest.shp\\roads.shp"
18 ret = processing.runalg('qgis:reprojectlayer', roads_shp_path, 'EPSG:3460',
19 output = ret['OUTPUT']
20 print output
21
```

19. Run the script via launch.bat and you will see the path to the newly created reprojected layer.

```
C:\Windows\system32\cmd.exe
Windows Live;C:\Program Files (<x86>)\Common Files\Microsoft Shared\Windows Live;C:\Program Files (<x86>)\Intel\iCLS Client\;C:\Program Files\Intel\iCLS Client\;C:\windows\system32;C:\windows;C:\windows\System32\Wbem;C:\windows\System32\Windows PowerShell\v1.0\;C:\Program Files\Intel\Intel(R) Management Engine Components\DA\;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT\;C:\Program Files (<x86>)\Intel\Intel(R) Management Engine Components\IPT\;C:\Program Files (<x86>)\Intel\OpenCL SDK\2.0\bin\x86;C:\Program Files (<x86>)\Intel\OpenCL SDK\2.0\bin\x64;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common Files\Intel\WirelessCommon\;C:\Program Files (<x86>)\Windows Live\Shared;C:\Program Files (<x86>)\QuickTime\QTSystem\;C:\Program Files (<x86>)\Skype\Phone\;C:\Program Files (<x86>)\Google\google_app_engine\

C:\Users\Ujaval\Desktop>SET PYTHONHOME=C:\OSGeo4W64\apps\Python27
C:\Users\Ujaval\Desktop>set PYTHONPATH=C:\OSGeo4W64\apps\qgis\python;
C:\Users\Ujaval\Desktop>REM Launch python job
C:\Users\Ujaval\Desktop>python c:\Users\Ujaval\Desktop\download_and_clean.py
C:\Users\Ujaval\AppData\Local\Temp\processing\48a32ea5978a49a4b7ee3a775c88a7bc\0
UTPUT.shp
C:\Users\Ujaval\Desktop>pause
Press any key to continue . . .
```

20. Now add the code for cleaning the topology. Since this is our final output, we will add the output file paths as the last 2 arguments for the `grass.v.clean` algorithm. If you left these blank, the output will be created in a temporary directory.

```
processing.runalg("grass:v.clean",
                  output,
                  1,
                  1,
                  None,
                  -1,
                  0.0001,
                  'C:\\\\Users\\\\Ujaval\\\\Desktop\\\\clean.shp',
                  'C:\\\\Users\\\\Ujaval\\\\Desktop\\\\errors.shp')
```

```

C:\Users\Ujaval\Desktop\download_and_clean.py - Notepad++
File Edit Search View Encoding Language Settings Macro Run Plugins Window ?
launch.bat download_and_clean.py

5     QgsApplication.setPrefixPath("C:\\OSGeo4W64\\apps\\qgis", True)
6     app = QgsApplication([], True)
7     QgsApplication.initQgis()
8
9     # Add the path to Processing framework
10    sys.path.append('c:\\Users\\Ujaval\\.qgis2\\python\\plugins')
11
12    # Import and initialize Processing framework
13    from processing.core.Processing import Processing
14    Processing.initialize()
15    import processing
16
17    # Reproject the Roads layer
18    roads_shp_path = "C:\\Users\\Ujaval\\Downloads\\fiji-latest.shp\\roads.shp"
19    ret = processing.runalg('qgis:reprojectlayer', roads_shp_path, 'EPSG:3460', None)
20    output = ret['OUTPUT']
21
22    # Clean the Roads layer
23    processing.runalg("grass:v.clean",
24                      output,
25                      1,
26                      1,
27                      None,
28                      -1,
29                      0.0001,
30                      'C:\\Users\\Ujaval\\Desktop\\clean.shp',
31                      'C:\\Users\\Ujaval\\Desktop\\errors.shp')
32    print 'Success'

```

Python file length: 1005 lines: 33 Ln:32 Col:16 Sel:0|0 Dos\Windows UTF-8 w/o BOM INS

21. Run the script and you will see 2 new shapefiles created on your Desktop. This completes the processing part of the script. Let's add the code to download the data from the original website and unzip it automatically. We will also store the path to the unzipped file in a variable that we can pass to the processing algorithm later. We will need to import some additional modules for doing this. (See the end of the tutorial for the full script with all the changes)

```

import os
import urllib
import zipfile
import tempfile

temp_dir = tempfile.mkdtemp()
download_url = 'http://download.geofabrik.de/australia-oceania/fiji-latest.shp.zip'

```

```

print 'Downloading file'
zip, headers = urllib.urlretrieve(download_url)
with zipfile.ZipFile(zip) as zf:
    files = zf.namelist()
    for filename in files:
        if 'roads' in filename:
            file_path = os.path.join(temp_dir, filename)
            f = open(file_path, 'wb')
            f.write(zf.read(filename))
            f.close()
        if filename == 'roads.shp':
            roads_shp_path = file_path

```

```

16
17 # Import and initialize Processing framework
18 from processing.core.Processing import Processing
19 Processing.initialize()
20 import processing
21
22 # Download and unzip the latest shapefile
23 temp_dir = tempfile.mkdtemp()
24 download_url = 'http://download.geofabrik.de/australia-oceania/fiji-latest.shp.zip'
25 print 'Downloading file'
26 zip, headers = urllib.urlretrieve(download_url)
27 with zipfile.ZipFile(zip) as zf:
28     files = zf.namelist()
29     for filename in files:
30         if 'roads' in filename:
31             file_path = os.path.join(temp_dir, filename)
32             f = open(file_path, 'wb')
33             f.write(zf.read(filename))
34             f.close()
35         if filename == 'roads.shp':
36             roads_shp_path = file_path
37
38 print 'Downloaded file to %s' % roads_shp_path
39
40 # Reproject the Roads layer
41 print 'Reprojecting the roads layer'
42
43 ret = processing.runalg('qgis:reprojectlayer', roads_shp_path, 'EPSG:3460', None)
44 output = rasterio.open(roads_shp_path)

```

Python file length:1718 lines:59 Ln:42 Col:1 Sel:0|0 Dos\Windows UTF-8 w/o BOM INS

22. Run the completed script. Everytime you run the script, a fresh copy of the data will be downloaded and processed.

```
C:\Windows\system32\cmd.exe
PowerShell\v1.0\;C:\Program Files\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\DAL;C:\Program Files (x86)\Intel\Intel(R) Management Engine Components\IPT;C:\Program Files (x86)\Intel\OpenCL SDK\2.0\bin\x86;C:\Program Files (x86)\Intel\OpenCL SDK\2.0\bin\x64;C:\Program Files\Intel\WiFi\bin\;C:\Program Files\Common Files\Intel\WirelessCommon\;C:\Program Files (x86)\Windows Live\Shared;C:\Program Files (x86)\QuickTime\QTSystem\;C:\Program Files (x86)\Skype\Phone\;C:\Program Files (x86)\Google\google_appengine\

C:\Users\Ujaval\Desktop>SET PYTHONHOME=C:\OSGeo4W64\apps\Python27
C:\Users\Ujaval\Desktop>set PYTHONPATH=C:\OSGeo4W64\apps\qgis\python;
C:\Users\Ujaval\Desktop>REM Launch python job

C:\Users\Ujaval\Desktop>python c:\Users\Ujaval\Desktop\download_and_clean.py
Downloading file
Downloaded file to c:\users\ujaval\appdata\local\temp\tmpvnus0n\roads.shp
Reprojecting the roads layer
Cleaning the roads layer
Success

C:\Users\Ujaval\Desktop>pause
Press any key to continue . . .
```

23. To automate running on this script on a daily basis, we can use the Task Scheduler in Windows. Launch the Task Scheduler and click Create Basic Task.

Note

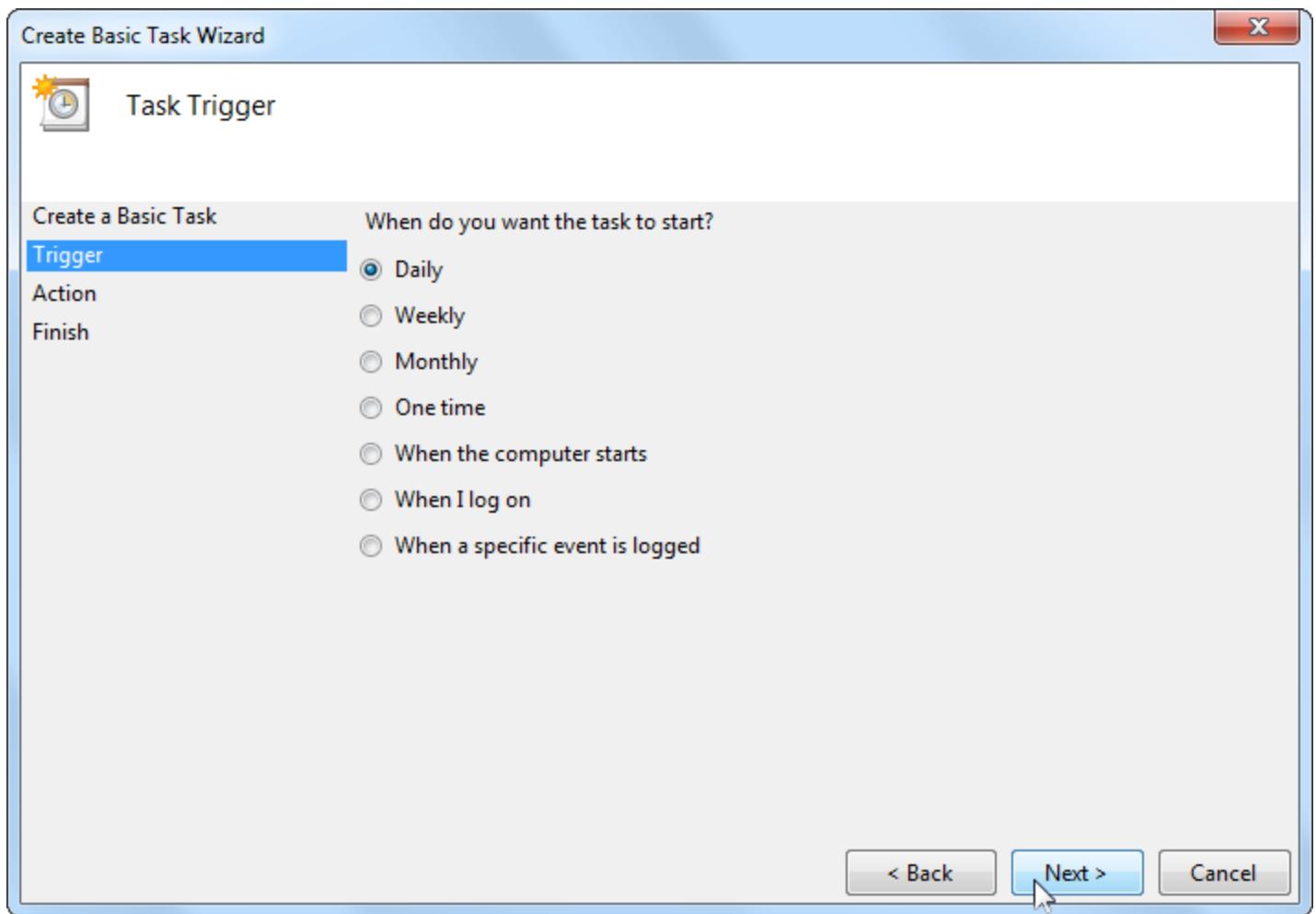
Linux and Mac users can use cron jobs to schedule tasks.



24. Name the task as Daily Download and Cleanup and click Next.



25. Select Daily as the Trigger and click Next



26. Select a time as per your liking and click Next.



27. Choose Start a program as the Action and click Next.



28. Click Browse and locate the `launch.bat` script. Click Next.



29. Click Finish at the last screen to schedule the task. Now the script will automatically launch at the specified time to give you a fresh copy of cleaned data everyday.



Below is the full download_and_clean.py script for your reference.

```
import sys
from qgis.core import *

import os
import urllib
import zipfile
import tempfile

# Initialize QGIS Application
QgsApplication.setPrefixPath("C:\\\\OSGeo4W64\\\\apps\\\\qgis", True)
app = QgsApplication([], True)
QgsApplication.initQgis()

# Add the path to Processing framework
sys.path.append('c:\\\\Users\\\\Ujava\\\\.qgis2\\\\python\\\\plugins')

# Import and initialize Processing framework
from processing.core.Processing import Processing
Processing.initialize()
import processing

# Download and unzip the latest shapefile
temp_dir = tempfile.mkdtemp()
```

```

download_url = 'http://download.geofabrik.de/australia-oceania/fiji-latest.shp.zip'
print 'Downloading file'
zip, headers = urllib.urlretrieve(download_url)
with zipfile.ZipFile(zip) as zf:
    files = zf.namelist()
    for filename in files:
        if 'roads' in filename:
            file_path = os.path.join(temp_dir, filename)
            f = open(file_path, 'wb')
            f.write(zf.read(filename))
            f.close()
            if filename == 'roads.shp':
                roads_shp_path = file_path

print 'Downloaded file to %s' % roads_shp_path

# Reproject the Roads layer
print 'Reprojecting the roads layer'

ret = processing.runalg('qgis:reprojectlayer', roads_shp_path, 'EPSG:3460', None)
output = ret['OUTPUT']

# Clean the Roads layer
print 'Cleaning the roads layer'

processing.runalg("grass:v.clean",
                 output,
                 1,
                 1,
                 None,
                 -1,
                 0.0001,
                 'C:\\\\Users\\\\Ujaval\\\\Desktop\\\\clean.shp',
                 'C:\\Users\\\\Ujaval\\\\Desktop\\\\errors.shp')
print 'Success'

```