# <https://github.com/Drozd363/ArcGIS_arcpy_S9_15...git>

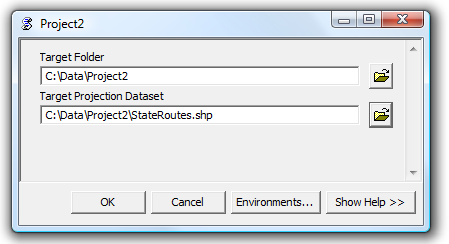
# P1: Інструмент пакетного перепроекціювання для векторних наборів даних

Нехай деякі ГІС відомства визначили одну, стандартну проекцію, яку використовують для підтримки їх джерел даних. Вихідні набори даних, однак, можуть бути отримані від третіх сторін в інших проекціях. Ці набори даних, необхідно перепроекціювати в стандартну проекцію відомства. Пакетне перепроекціювання, або перепроектування багатьох наборів даних одночасно, є завданням що добре підходять для написання скриптів.

У цьому проекті ви будете практикуватись у Python, написавши скрипт, перепроекціювання векторих наборів даних (datasets) у папці. З цього скрипта, ви потім створити скрипт інструмент, яким можна легко поділитися з іншими.

Інструмент, що ви напишете повинен виглядати як на картинці нижче. Він складається з двох вхідних параметрів і вихідних параметрів немає. Дві вхідні параметри:

1. Папка на диску, що містить набори векторних даних, що повинні отримати нову проекцію.
2. Шлях до векторних даних, просторова прив’язка яких буде використовуватися при перепроекціюванні. Наприклад, якщо ви хочете перепроекціювати в NAD 1983 UTM Zone 10, ви повинні знайти набір даних, що вже в NAD 1983 UTM Zone 10. Це може бути один з наборів даних у папці, яку ви надали в першому параметрі, або він може існувати в іншому місці на диску.

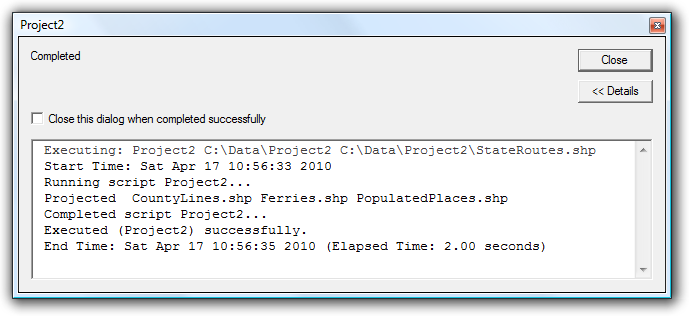


Запуск програми викликає перепроекціювання даних розміщених на диску в цільовій папці.

**Вимоги**

Ваш скрипт:

* Повинен перепроектовувати набори shapefile у папці відповідно з проекцією цільового набору даних.
* Повинен додавати "\_projected" в кінець імен кожного з перепроектованих наборів даних. Наприклад: CityBoundaries\_projected.shp.
* Повинен пропустити набори даних, які вже знаходяться в цільовій проекції.
* Повинен повідомляти geoprocessing message, які набори даних отримали нову проекцію. У цьому повідомленні, набір даних імена можуть бути розділені пропусками. У повідомленні, не включати набори даних, які були пропущені, тому що вони вже були в цільовій проекції. Зверніть увагу на приклад цього типу користувацького повідомлення нижче в рядку "Projected . . .":



* Не повинен містити жорстко закодованих значень, таких як набір імен даних, імена шляхів, або імена проекцій.
* Повинен бути доступним в якості скриптового інструменту, який може бути легко запущеним з ArcToolbox кимось, хто не знає скрипти.

Ви не зобов'язані виконувати перетворення датумів (datum) у своєму скрипті. Передбачається, що кожен набір даних в папці використовує той же самий датум, хоча набори даних можуть бути в різних проекціях.

**Дані для роботи**

У вашій папці даних **Lesson 2** є набір shapefiles для вас, щоб працювати з ними в цьому проекті. Ці shapefiles були отримані з [Washington State Department of Transportation GeoData Distribution Catalog](http://www.wsdot.wa.gov/mapsdata/GeoDataCatalog/default.htm), і вони представляють різні географічні особливості навколо штату Вашингтон. Для цілей цього проекту, я поставив ці набори даних у різних проекціях. Ці проекції в одні й ті ж системі datum (NAD 83), так що вам не доведеться мати справу з datum перетвореннями.

Набори даних і їх оригінальні проекції:

* **CityBoundaries** and **StateRoutes** - NAD\_1983\_StatePlane\_Washington\_South\_FIPS\_4602
* **CountyLines** - NAD\_1983\_UTM\_Zone\_10N
* **Ferries** - USA\_Contiguous\_Lambert\_Conformal\_Conic
* **PopulatedPlaces** - GCS\_NorthAmerican\_1983

**Очікується**

Очікувані результати для цього проекту полягають у наступному:

* Ресурс .py файл, що містить скрипт
* .tbx файл, що містить ваш скрипт

**Hint**

Наступні поради можуть допомогти вам у цьому проекті:

* Не використовуйте інструмент Esri Batch Project tool у цьому проекті. По суті, ви зобов'язані зробити ваш власний варіант інструменту пакетного перепроекціювання у цьому проекті, запустивши проект інструменту всередині циклу. Ваш інструмент буде простіше у використанні, тому що він налаштований під задачу.
* Є багато способів, щоб вставити "\_projected" в ім'я набору даних, але вам, можливо, буде корисно почати з тимчасового видалення ".shp" і додавання його знову пізніше. Для видалення ".shp" ви можете використовувати синтаксис на зразок цього:

rootName = ""

if fc.endswith(".shp"):

rootName = fc[:-4]

У наведеному вище коді, fc ім'я вашого shapefile з .shp дозширеням, і fc[:-4] усуває останні чотири символи.

* Щоб перевірити, чи є набір даних вже в цільової проекції, ви повинні будете отримати Spatial Reference (просторову прив’язку) на об'єкт для кожного набору даних (набору даних для проектування і цільового набору даних). Потім вам потрібно буде порівняти просторові імена посилань з цих двох наборів даних. Будьте впевнені у властивості Name просторових посилань; не зрівнюйте просторові посилання об'єктів до них. Це тому, що ви можете мати два просторових посилання для об'єктів, які відрізняються іменами (і, таким чином, "не дорівнюють"), але мають однакове ім'я властивості.

Ви повинні створити рядок, подібно до цього:

if fcSR.Name != targetSR.Name:

де fcSR є просторовою прив'язкою класу об'єктів для проектування і targetSR є цільовою просторовою прив'язкою отриманою з цільового shapefile.

* Якщо ви хочете показати всі повідомлення для кожного запуску вашого інструменту, додайте рядок: arcpy.AddMessage (arcpy.GetMessages ()) відразу після рядка, де ви запустите проект інструменту. Кожен раз, коли цикл виконується, це додасть повідомлення для поточного виконання проекту інструменту у вікні результатів. Якщо ви чекаєте, щоб додати цю лінію до кінця сценарію, то ви можете отримати повідомлення тільки для останнього запуску інструменту, тому дуже важливо, щоб поставити цю лінію всередині циклу. Пам'ятайте, що в той час коли ви вперше написали скрипт, то можна використовувати оператори друку для налагодження, потім перемкнутися на arcpy.AddMessage (), коли ви переконалися, що ваш скрипт працює, і ви готові зробити скрипт інструмент.

Всі результати повинні бути портованими та представлені на Git. Крім того не забувайте, що ця робота є командною, тож **використовуйте** Git.

import arcpy  
arcpy.env.overwriteOutput = True  
  
arcpy.env.workspace = arcpy.GetParameterAsText(0)  
input = arcpy.GetParameterAsText(1)  
#arcpy.env.workspace = r"E:\programin\semestr2\samrob\S10\_GIS\_FILE"  
#input = "CountyLines.shp"  
  
inputSR = arcpy.Describe(input).SpatialReference  
inputSRname = inputSR.name  
arcpy.AddMessage("Coordinate systems will be changed to {}".format(inputSRname))  
listfc = arcpy.ListFeatureClasses()  
for fclas in listfc:  
 clasSR = arcpy.Describe(fclas).SpatialReference  
 clasSRname = clasSR.name  
 if clasSRname != inputSRname:  
 output = fclas[:-4] + "\_projected.shp"  
 arcpy.Project\_management(fclas, output, inputSR)  
 arcpy.AddMessage("The coordinate system file '{}' has been changed to {}".format(fclas, str(inputSRname)))  
 else:  
 arcpy.AddMessage("The coordinate system of the file '{}' does not need to be changed".format(fclas))