How to setup the Profile geoprocessing service using ArcGIS Pro

Last update: 10/27/2023

Introduction

The ArcGIS Online elevation [Profile service](#) is published from a Profile script tool. With proper configuration, you can publish the same Profile tool to your in-house server and set up a Profile service that operates on your own elevation data. This document provides guidance on how to configure the Profile script tool and publish it as a geoprocessing service on ArcGIS Enterprise.

Data Requirement

Required Datasets

The Profile script tool requires Digital Elevation Model (DEM) data and a few other supporting components. The following is a full list of the required datasets.

1. The **DEM** data, which provides the elevation values for the profile. In this application, the DEM data needs to be stored in one or more mosaic datasets (each DEM resolution requires one mosaic dataset).

2. A **DEM resolution boundary** feature class, which consists of several polygon features. Each polygon represents the coverage area of the corresponding DEM resolution. The feature class must contain a field called "res" that records the resolution of the coorresponding DEM, which must be of long integer type. The following screen capture is an example of the attribute table of a DEM boundary feature class.

	OBJECTID *	Shape *	res	Shape_Length	Shape_Area
1	1	Polygon	90	89257.073303	519010849.902829
2	3	Polygon	10	49595.224	147787856.334637
3	4	Polygon	30	64221.612	219603487.213865
	Click to add new row.				

3. An **input line schema**. This is a polyline feature class used by the Profile tool as the feature schema for the input line feature parameter.

Folder structure and data organization

Under the Profile folder, the python toolbox Profile Tool Pro.pyt contains the Profile tool. The ProfileData folder contains two file geodatabases (dembnd.gdb and demdata.gdb). The demdata.gdb stores the DEM mosaic datasets, while the dembnd.gdb stores the DEM resolution boundary feature class and the input profile line schema. The following screen capture shows the contents of the Profile folder.
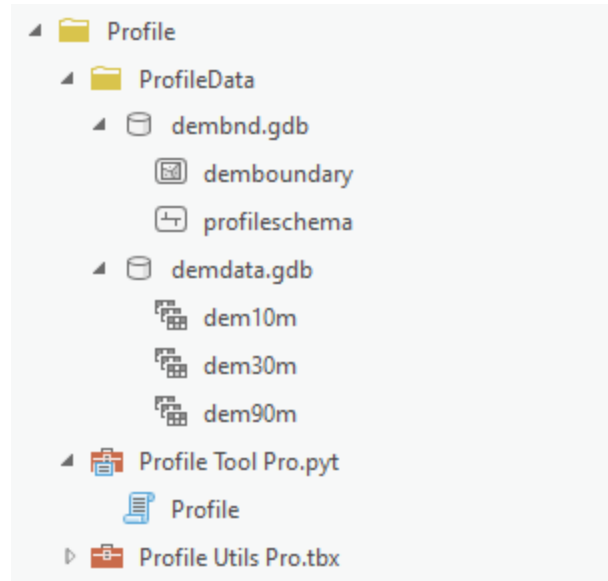
Create the DEM mosaic datasets

A mosaic dataset is used to integrate multiple raster datasets into one dataset, without creating a large raster on disk. In this Profile service, mosaic datasets are used to manage DEM's for each of the resolutions.

For each DEM resolution, a new mosaic dataset needs to be created using the Create Mosaic Dataset geoprocessing tool. Then the individual DEM images need to be added to the mosaic dataset using the Add Rasters to Mosaic Dataset tool.

Create the DEM boundary feature class

The Profile Utils Pro.tbx toolbox provides a utility tool 'Create DEM Boundary' to facilitate the creation of the DEM boundary feature class from a given mosaic dataset. The steps are:

1. In Pro, open the Create DEM Boundary tool under the Profile Utils Pro.tbx. For the first parameter, specify an input DEM mosaic dataset. The second parameter is DEM cell size, which should be specified as an integer value in meters. The output can be saved to a temporary workspace.

2. If there are multiple DEM resolutions, repeat step 1 for each resolution. If there is only one DEM, then go to step 3.

3. Use the Merge geoprocessing tool (under Data Management -> General in ArcToolbox) to merge all the individual DEM boundary feature classes into one feature class. Save the output as 'demboundary' to dembnd.gdb.
    Note that if there is only one resolution, you can just copy the DEM boundary feature class from step 1 to dembnd.gdb and name it 'demboundary', using the Copy Features tool.

Create the input line feature schema

The input line feature schema is a polyline feature class. You can create such a feature class using the Pro editing tools, or use an existing feature class on disk. This feature class should be copied to the dembnd.gdb.

Updating the Profile Tool Script

Once the data is ready, you can update the Profile tool script to point to the data. The implementation of the Profile tool is found in the python toolbox 'Profile Tool Pro.pyt'.

Open 'Profile Tool Pro.pyt' in a python script editor, find the __init__ function of the class Profile and update the following variables.

1. **Define maximum number of elevation values for each profile**
    In line 39, the maximum number of elevation values on the output profile is set. By default, it is 2000. You may change this to any integer value or accept the default.

```
36          #-----------------------------------------------------
37          # Maximum number of vertices
38          #-----------------------------------------------------
39          self.maxNumVertices = 2000
```

2. **Specify the DEM boundary feature class**
    The DEM boundary feature class is set in lines 43 and 44. In line 43, the path to the dembnd.gdb file geodatabase is specified. In line 44, the full pathname of the DEM boundary feature class is constructed.

```
40          #-----------------------------------------------------
41          # DEM boundary layer
42          #-----------------------------------------------------
43          boundaryGdbPath = r'C:\Profile\ProfileData\dembnd.gdb'
44          boundaryLayer1 = os.path.join(boundaryGdbPath, "demboundary")
45          if False:
46              arcpy.Describe(boundaryLayer1)
47          self.demBoundary = boundaryLayer1
```

3. **Specify the input line schema feature class**
    The input line feature schema is specified in line 51. The default feature class name is 'profileschema'. Depending on the name used when you create this feature class, a different feature class name can be specified.

```
48          #-----------------------------------------------------
49          # Profile schema feature class
50          #-----------------------------------------------------
51          profileSchm1 = os.path.join(boundaryGdbPath, "profileschema")
52          self.profileSchema = profileSchm1
```

4. **Define the DEM resolutions dropdown**
    The DEM Resolution dropdown (of the 3rd tool parameter) is defined as a dictionary in line 56. Each resolution is specified as a key value pair. The key is a descriptive name for the resolution, such as '90m'. The value is an integer (as a string) representing the resolution of the DEM, such as '90'. The default DEM resolution is defined in line 57.

```
53          #----------------------------------------------------
54          # DEM resolution dictionary
55          #----------------------------------------------------
56          self.dictDEMResolutions = {"90m":"90", "30m":"30", "10m":"10"}
57          self.defaultDEMResolution = '90'
```

5. **Specify DEM data layers**
   A. In line 61, the path to the demdata.gdb is specified. In lines 62, 63, etc, the full pathnames of each DEM mosaic dataset are created. Depending on the number of DEM resolutions you may have, add or remove lines here.

```
58          #----------------------------------------------------
59          # DEM data layers
60          #----------------------------------------------------
61          mosaicGdbPath = r"C:\Profile\ProfileData\demdata.gdb"
62          demLayer1 = os.path.join(mosaicGdbPath, "dem90m")
63          demLayer2 = os.path.join(mosaicGdbPath, "dem30m")
64          demLayer3 = os.path.join(mosaicGdbPath, "dem10m")
```

   B. Wrap each DEM layer variable in an arcpy.Describe statement. These statements are never executed, but they are needed for the publisher to detect the data layers during publishing. Depending on the number of DEM resolutions you may have, add or remove lines here.

```
65          #----------------------------------------------------
66          # Wrap each variable in an arcpy.Describe statement
67          #----------------------------------------------------
68          if False:
69              arcpy.Describe(demLayer1)
70              arcpy.Describe(demLayer2)
71              arcpy.Describe(demLayer3)
```

   C. In lines 75, 76, etc, the DEM data layers are associated with their resolutions in a dictionary as key value pairs. The keys are the resolutions, the values are the DEM data layers. Depending on the number of DEM resolutions you may have, add or remove lines here.

```
72          #----------------------------------------------------
73          # Update the DEM layers dictionary
74          #----------------------------------------------------
75          self.dictDEMs = {"90":demLayer1,
76                           "30":demLayer2,
77                           "10":demLayer3}
```

Note:
   You can make the following changes if your DEM is in feet, and you want to represent DEM resolutions and M values in feet:
   1. Populate the DEM resolution parameter dropdown using values in feet by updating self.dictDEMResolutions accordingly.
   2. Change the computation of M value to feet by updating the glen_field2 field calculation (line 464) in the execute method.

Publishing the Profile Service to ArcGIS Enterprise

Once the data and script are ready, the Profile script tool can be published as a service.

Register the profile data folder on ArcGIS server
Before publishing the Profile service, the Profile data folder needs to be registered on the server machine, so that no large amount of data needs to be copied by the publisher in later steps.

1. Copy the data folder 'ProfileData' to the hosting server of your ArcGIS Enterprise setup. It is recommended to put the data path on the Server same as the data path on the publisher machine (such as C:\Profile\ProfileData for both).

2. In a web browser, open the ArcGIS server manager and sign in with an administrator or publisher account.

3. Go to Site → Data Stores and select Register Folder to register the ProfileData server folder.

For more help on how to register a folder on ArcGIS server, visit Register data with ArcGIS Server using Server Manager .

Publish the Profile service to ArcGIS Enterprise
1. Open ArcGIS Pro, ensure that the Spatial Analyst and the 3D Analyst extension licenses are enabled. Make a publisher or administrator connection to the ArcGIS Enterprise where the Profile service will be hosted.

2. In Pro catalog window, connect to the folder where the Profile Tool Pro.pyt toolbox file is located. Open the Profile tool and run it once.

3. Open the Geoprocessing History window (under the Analysis ribbon), if it is not already open.

4. In the history window, find the history of the Profile tool run. Right-click the history entry and click Share As→Share Web Tool.

5. Follow the Share As Web Tool wizard to publish the Profile tool to your ArcGIS Enterprise. Ensure the following configurations are performed correctly:
   i. The Reference registered data option should be checked on the General tab.
   ii. Set the sharing level to ArcGIS Enterprise on the General tab.
   iii. On the Contents tab, click the Pencil control 🖉 beside the Profile tool to activate the Profile Properties dialog. Uncheck the Add optional output feature service parameter option (as shown in the screen capture below).

6. Click the Publish button to start publishing. If there are any errors returned, fix them and click the Publish button again to continue.

For more help on publishing a geoprocessing service, visit [Quick tour of authoring and sharing geoprocessing services](#).