



Textured Painted Relief Tool 1.0

Tool documentation

Dušan Maďar

01.02.2013

This documentation contains information about content of the installation directory as well as descriptions of included scripts, configuration and validation file and usage requirements.

Table of contents

1.	Basic description	2
2.	System requirements	2
3.	Installation directory content	2
4.	Addin directory content	3
5.	Addin\Install directory content	3
6.	Configuration file	4
6.1.	Textures settings	4
6.1.1.	Randomness effect	8
6.2.	Landuses settings	8
7.	User interface	9
8.	How to run TPR in ArcMap	10

1. Basic description

Purpose of this tool is to create a Textured Painted Relief based on user inputs and a configuration file, or in other words, to automatize the Textured Painted Relief generating process. Expected user inputs are preprocessed raster or vector terrain and landuse layers (ESRI Grid or TIN for terrain; ESRI Grid or ShapeFile for layers). TPRT was developed as a diploma thesis on Institute of geoinformatics, VŠB – Technical University of Ostrava, Czech Republic. It is implemented in Python programming language, based on ArcGIS site package ArcPy. TPR also uses a XML configuration file and a XSD schema to validate it. GUI of the tool is implemented using wxPython framework. The tool can be used from ArcGIS (ArcMap) interface as an Add-in or outside of ArcGIS as a standalone script (ArcPy is still necessary though).

TPR tool is inspired by Jeffery Nighberts (father of the Textured Painted Relief idea) work and projects. You can find some of the original projects on the following websites:

- [Creating Beautiful "Painted Relief" Using Arc/Info](#)
- [Characterizing Landscapes for Visualization Through "Bump Mapping" and Spatial Analyst](#)
- [A Concise History of Bump Mapping](#)
- [Using ArcGIS to Apply Textures and Materials to Relief Backdrops in Cartographic Presentations](#)
- [Symbolizing the Bump Map](#)

2. System requirements

- ArcGIS Desktop 10.1
 - Spatial analyst
 - 3D Analyst
- Python 2.7
 - lxml 2.3.6
 - wxPython 2.8

3. Installation directory content

\Addin	Tool scripts, images, XML schema.
\Documentation	Brief tool description, XSD validation schema and an example of XML configuration file, Class diagram, Cartographic model.
\InstallPythonModules	Necessary Python modules.
\TPR.esriaddin	TPR tool ESRI Addin.

4. Addin directory content

\Install

The Python project used for the implementation of the Addin + all UI images for the project (icons, images for buttons, etc).

\[Content_Types].xml

XML describing project file types.

\config.xml

Addin configuration file (defining the static add-in properties).

\makeaddin.py

A script that will create an .esriaddin file out of this project, suitable for sharing or deployment.

\README.txt

Default readme file made by ArcGIS Desktop Python AddIn Wizard.

5. Addin\Install directory content

_addButton.png

Basic add data button.

_addButton_D.png

Add data button for data which are encapsulated in a directory (e.g. ESRI Grid or TIN).

_saveButton.png

Save data button.

_tpr.ico

TPR window icon.

_tpr.png

TPR ArcMap toolbar icon.

\settings.xsd

TPR XSD validation schema.

\TexturedPaintedRelief_addin.py

TPR ArcMap Addin. Enables to run the tool from ArcMap interface.

\TexturedPaintedRelief_interface.py

TPR graphic user interface. **Can be used outside of the ArcMap interface – as a standalone script.**

\TexturedPaintedRelief_logic.py

TPR logic. Implements all necessary functionality – environments settings; textures, bumpmap, landuse and hillshade creation; generating the TPR. **Can be used as a standalone script (data definition necessary).**

\TexturedPaintedRelief_validator.py

TPR user data controller. Tests and validates user data – configuration file, textures specification, hillshading options and directory write access.

6. Configuration file

Both textures and landuses settings are stored in a XML file which is validated against a XSD file (settings.xsd). This XSD (XML Schema Definition) **defines configuration file structure** – how each of the supported textures and landuses **must be defined**. You need to set arguments in the **right order, using appropriate data types** and **correct XML elements**.

Referencing a XML configuration file is the very first step when using the TPRT tool. Specified configuration file is validated against XML schema (settings.xsd) right afterwards. TPR can't be used if the validation process is unsuccessful (combo box choices won't be updated), so **writing a good configuration file is essential**.

6.1. Textures settings

Squares texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) randomness	Squares distribution type. For example 0 = regular, 1 = random, 10 = very random, ...
(integer) density	Squares centroids distance in map units. STRONGLY affected by randomness.
(integer) size	Squares side in map units.
(integer) height	Square height in map units.

XML example (if map units are meters: squares1 – 20m side, 40m high, 13m between each square, randomness = 5):

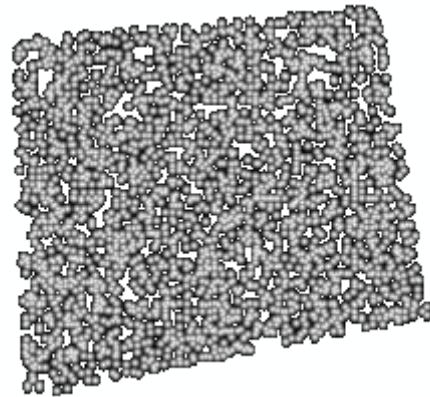
```
<squares>
  <name>squares1</name>
  <randomness>5</randomness>
  <density>13</density>
  <size>20</size>
  <height>40</height>
</squares>
```



Cones texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) randomness	Cones distribution type. For example 0 = regular, 1 = random, 10 = very random, ...
(integer) density	Cones centroids distance in map units. STRONGLY affected by randomness.
(integer) size	Cones diameter in map units.
(integer) height	Cones height in map units.

XML example (if map units are meters: cones1 – 20m diameter, 40m high, 13m between each square, randomness = 2):

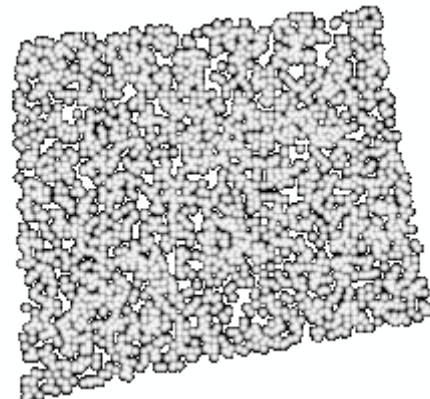
```
<cones>
  <name>cones1</name>
  <randomness>2</randomness>
  <density>13</density>
  <size>20</size>
  <height>40</height>
</cones>
```



Spheres texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) randomness	Spheres distribution type. For example 0 = regular, 1 = random, 10 = very random, ...
(integer) density	Spheres centroids distance in map units. STRONGLY affected by randomness.
(integer) size	Spheres diameter in map units.

XML example (if map units are meters: spheres1 – 20m diameter (20 high), 13m between each square, randomness = 2):

```
<spheres>
  <name>spheres1</name>
  <randomness>2</randomness>
  <density>13</density>
  <size>20</size>
</spheres>
```

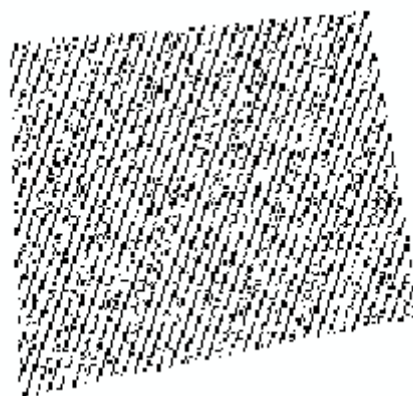


Plough texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) angle	Ploughing orientation in degrees. 0 degrees means ploughing is parallel with Y-axis Values in interval <0,180>.
(integer) interval	Distance between individual "lines".

1

XML example (if map units are meters: plough1 – 20° lines orientation, 20m between lines):

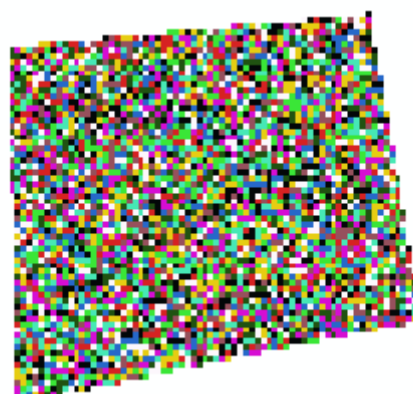
```
<plough>
  <name>plough1</name>
  <angle>20</angle>
  <interval>20</interval>
</plough>
```



Noise texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) minimum	Minimum texture value (pixel "height").
(integer) maximum	Maximum texture value (pixel "height").

XML example (if map units are meters: noise1 – 20m wide line, 20m high line):

```
<noise>
  <name>noise1</name>
  <minimum>1</minimum>
  <maximum>10</maximum>
</noise>
```



Lines texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) width	Line width in map units.
(integer) height	Line height in map units.

XML example (if map units are meters: lines1 – 20m wide line, 20m high line):

```
<lines>
  <name>lines1</name>
  <width>20</width>
  <height>1</height>
</lines>
```

Lines texture is designed and can be used **only with line ShapeFiles**. For example if you have a line Shapefile representing roads and you want to make these roads more visible in the output – use Lines texture with appropriate width and height.

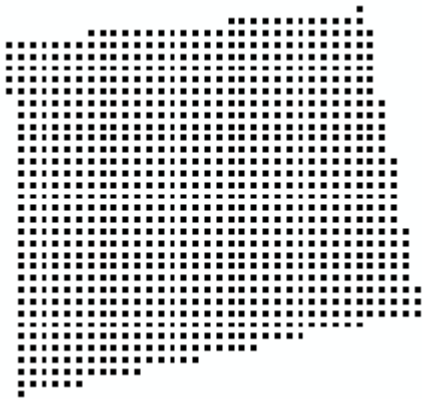
Null texture	
(string) name	Name of the texture which will appear in the Texture combo box list.
(integer) value	Output raster value.

XML example (if map units are meters: null1 – layer area will be 1m cut into the terrain):

```
<null>
  <name>null1</name>
  <value>-1</value>
</null>
```

When Null texture is used **no specific texture is generated**. This type of texture just takes the referenced layer and reclassifies its values to entered value. For example if you use a raster that represents roads (road with is already defined by raster) and you just want to add it to the output, use Null texture. Or if you have a polygon ShapeFile representing river, use Null texture with appropriate value as well.

6.1.1. Randomness effect



Randomness = 0



Randomness = 2



Randomness = 5

(Squares texture – density = 20, size = 10, height = 40)

6.2. Landuses settings

As well as textures, landuse setting are also stored in a XML configuration file, but landuse settings are far simpler. You just need to specify a landuse name and a z-index value.

```
...  
<landuse name="Road">  
  <z-index>900</z-index>  
</landuse>  
  
<landuse name="Water">  
  <z-index>800</z-index>  
</landuse>  
...
```

Z-index is a **unique number** used to get landuses in to the correct order. **The higher the value of z-index, the higher the referenced landuse will be placed.** In our example, landuse "Road" will be above landuse "River" (e.g. bridge is above a river).

7. User interface

The screenshot shows the 'Textured Painted Relief Tool' window. It has a title bar with standard window controls. The main area is divided into several sections:

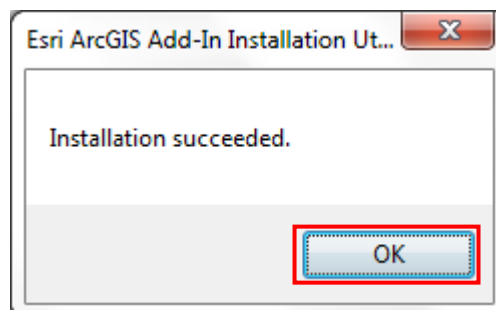
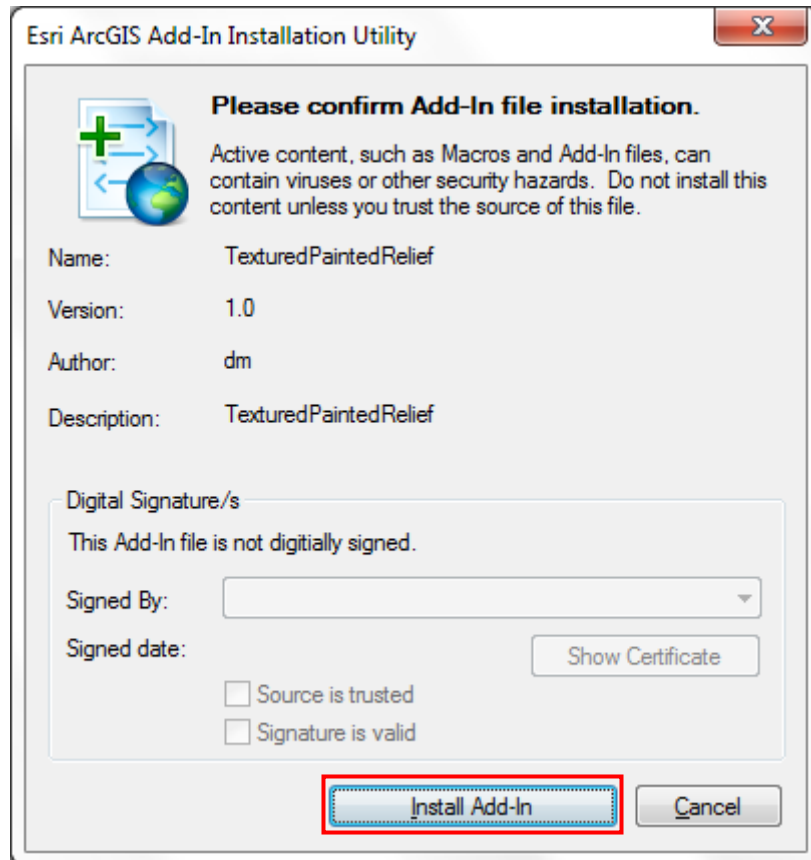
- Configuration 1**: A text input field with a green '+' button.
- Terrain 2**: A text input field with a green '+' button.
- Textures 3**: A table with columns for Layer, Texture, LandUse, and Color. It has three rows, each with a green '+' button (3.1), a blue '+' button (3.2), a 'Select Texture' dropdown (3.3), a 'Select LandUse' dropdown (3.4), and a color selection field (3.5). An 'Add row' button is at the bottom right of the table.
- Hillshading options 4**: A section with input fields for Azimuth (315), Altitude (45), Z factor (1.00), a dropdown for Shadows (No), a text field for Cell Size (10), and a color selection field.
- Output 5**: A text input field with a green '+' button.
- TPR Tool**: A sidebar on the right containing text: 'This tool creates a textured painted relief based on a XML configuration file and user inputs.', 'Note 1: Texture and LandUse combobox choices will be available after setting a valid configuration file.', and 'Note 2: To show specific help just click the field you want to know more about.' Below this is a red number '7'.
- 6**: A status bar at the bottom left.

Buttons 'Ok' and 'Cancel' are at the bottom center.

1. Configuration file section
2. Terrain section
3. Textures section
 - 3.0. Select layer to be texturized
 - 3.1. Select ESRI ShapeFile button
 - 3.2. Select ESRI Grid button
 - 3.3. Select Texture
 - 3.4. Select Landuse
 - 3.5. Select Texture Color
4. Hillshading section
5. Output section
6. Statusbar – all tool messages will be displayed here
7. Help section – help will be displayed here

8. How to run TPR in ArcMap

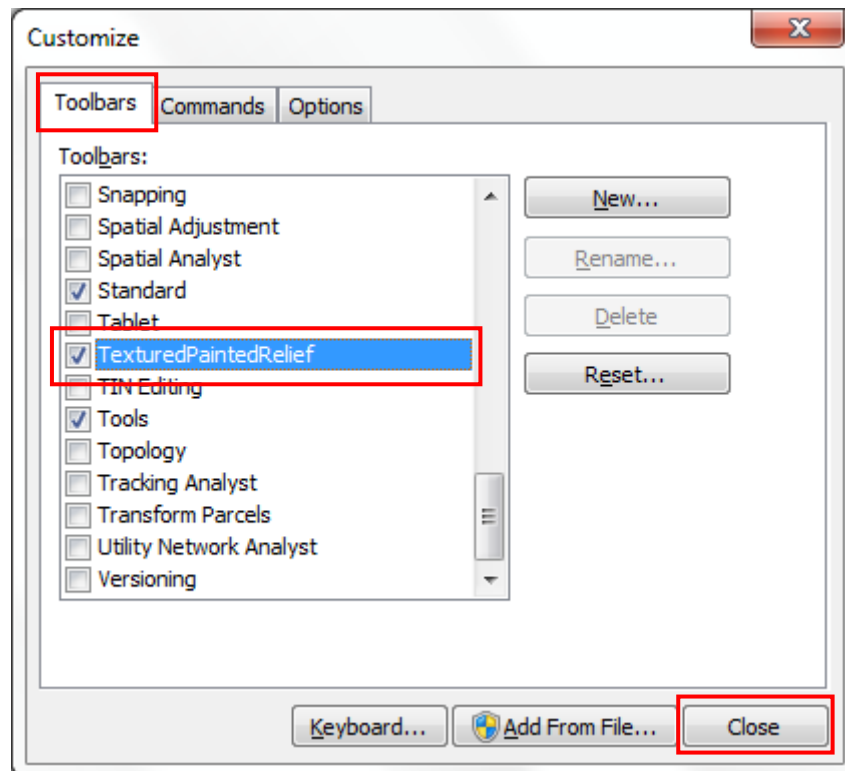
1. Just double-click the “TPR.esriaddin”, click “Install Add-in”, click OK



2. Run ArcMap
3. Click on “Toolbar Options/Customize...”



4. Scroll down to “TexturedPaintedRelief” and check it, click Close



5. Now you have a TPR icon on your toolbar, just click it and use the tool



More on this subject:

- [Sharing and installing add-ins](#)
- [Adding and removing tools on menus and toolbars](#)