**CACIE Tool #07.4** – ***Rotate Slice Tool (tecplotauto\_rotate\_slice.py)***

**Version** **1.0**

**QA**: **TEST** or **NA** or **QA**

1. **Description and Purpose**

One or two paragraphs describing the tool’s function and purpose.

The Rotate Slice tool automates the Tecplot® software package to generate the following:

* Orientations of a geologic cross section (.lay file) in an east to west, a north to south, a west to east and a south to north direction. Each orientation is saved as a .png file.
* Slices of the same geologic cross section. The orientation of the cross section and the coordinates of the slices are pre-defined in a modelname\_input\_sij\_pr2.dat file. Each sliced cross section is saved to a .png file

1. **Functional Requirements**

The functional requirements of the tool will be documented in this section. Each requirement will have an ID, such as: FR-N, where N starts at 1 and increments for each Functional Requirement. Each of the Functional Requirement IDs will have a corresponding test ID listed in the RTM.

FR-1: Connect to the tecplot software

FR-2: Load layout file (argument from the .bat file)

FR-3: Rotate cross section to generate west (60º), north (150º), south (330º), east (240º), and orientations and save to .png file

FR-4: From modelname\_input\_sij\_p2r.dat file, parse slicing direction and slice coordinates

FR-5: Generate corresponding cross section slices

FR-6: Save each cross section slice as a .png file

1. **Software Requirements Specifications**

The software requirements specification of the tool will be documented in this section.

Python 3.6

Python library dependencies:  
tecplot   
tecplot.exception   
tecplot.constant  
os  
logging  
argparse  
sys

1. **Software Design Description**

The software design description of the tool will be documented in this section. The results of a Code Walkthrough with an independent third party will be summarized in this section.

Arguments:  
modelname-d-CCCCCC.lay [d=directional orientation of front face of cross section, CCCCCC=coordinate]

Attribute-only object Class: TpLayoutDetail

Functions:

create\_logger(path\_filename,log\_level='0')

rotate\_360degrees(theta\_dict, frame, layout\_deets, directory)  
get\_layout()  
get\_slices (path\_file)  
get\_tpdetails (frame, model\_name)  
get\_txtcoords(theta\_dict, front\_face, frame)

verify\_dir(directory)  
get\_datfile(model)  
save\_png(path\_filename)  
gen\_slice\_pngs(slices\_dict, path, frame, theta\_dict, layout\_deets)

Outputs:

png files

.bat file: run\_tecplotauto\_rotate\_slice.bat:

* python ../../../tools/tecplotauto/tecplotauto\_rotate\_slice.py modelname-d-CCCCCC.lay

1. **Requirements Traceability Matrix**

A requirements traceability matrix for the tool will be documented in this section. At a minimum, the matrix will include IDs of: Functional Requirements and the corresponding Acceptance Test, along with an indication of the test result (Pass/Fail).

Table 1 presents the requirements traceability matrix for the Turn\_ChangeLabels\_Slice\_Save tool.

| **Table 1. Turn\_ChangeLabels\_Slice\_Save Tool Requirements Traceability Matrix** | | |
| --- | --- | --- |
| **Functional Requirement** | **Acceptance Test** | **Test Result  (Pass/Fail)** |
| FR-1 | ATC-1 |  |
| FR-2 | ATC-2 |  |
| FR-3 | ATC-3 |  |
| FR-4 | ATC-4 |  |
| FR-5 | ATC-5 |  |

1. **Test Plan and Cases**

The test plan for the tool will be documented in this section. Each test will have a unique ID and criteria for determining if the test result is pass or fail. The TEST ID will be referenced in the RTM and ATR. An installation test, labeled **IT-1**, will be used by the Tool Runner to confirm the version of the tool being used is running correctly before launching it with the user’s parameters.

The Unit Testing done on the tool will be documented here, also.

The test plan for the Rotate Slice tool is as follows.

| **Table 2. Rotate Slice Tool Test Plan** | | |
| --- | --- | --- |
| **TEST ID** | **Test Case** | **Test Result (Pass/Fail)** |
| IT-1 | Installation Test |  |
| ATC-1 | Confirm script executes (script will terminate if TecPlot connection is not made) |  |
| ATC-2 | Check Tecplot interface to confirm loaded layout filename matches .bat argument filename |  |
| ATC-3 | * In saved .png file with the same orientation as the .lay file, verify front face label and lower right coordinate value corresponds to .lay filename   Record coordinate ranges of cross section (Easting1 to Easting2 and Northing1 to Northing2) for further acceptance testing [ATC-3]. |  |
| * For four cross section .png files, verify that front face label and the lower right coordinate value corresponds to filename (modelname-d-CCCCCC.lay). |  |
| For the cross section orientation .png files, verify the following as applicable:   * North Orientation .png: Front Label = North and Right Label = West; North Face = Easting2 to Easting1, West Face = Northing2 to Northing1 * West Orientation .png: Front Label = West and Right Label = South; West Face = Northing2 to Northing1, South Face = Easting1 to Easting2 * South Orientation .png: Front Label = South and Right Label = East; South Face = Easting1 to Easting2, West Face = Northing2 to Northing1 * East Orientation .png: Front Label = East and Right Label = North; East Face = Northing1 to Northing2, North Face = Easting2 to Easting1 |  |
| ATC-4 | For one .png file in each slicing direction, verify that the following are consistent with the specifications in the .dat file:   * slicing direction, * cross section orientation * coordinate of slice |  |
| ATC-5 | Verify that a cross section slice .png file was generated for each slicing direction and coordinate specified in modelname\_input+sij\_p2r.dat (open .dat file in text editor of choice) |  |

1. **Acceptance Test Report**

The test report will state whether the tool is qualified for use, summarize test case results, and report all resolved incidents and resolution of unresolved incidents.

1. **User Guide**

A guide for using the tool will be documented in this section.

With Tecplot software open and Scripting🡪PyTecplot Connections set to “Listen to localhost only”:

Generating slices at a given distance in the North-to-South and West-to-East directions:

1. A note before running the script:
   * 1. The python script below will use the ***[ModelName]\_input\_sij\_p24.dat*** file for the slice spacing. By default the ***[ModelName]\_input\_sij\_p24.dat*** file is 100 m sliced. If you choose to have a finer slicing you will need to modify they ***[ModelName]\_input\_sij\_p24.dat*** file by adding lines and values in between the highest and lowest ends.
2. Copy the ***run\_tecplotauto\_rotate\_slice.bat*** script from the CAVE/sara-sandbox/shells/ directory to your CAVE/sara-sandbox/testing-b3pond/ss/plts/ directory
3. Open the ***run\_tecplotauto\_rotate\_slice.bat*** script in your preferred text editor.
4. Change the layout filename as appropriate to correspond to the model being run.
5. If you want a different slice distance and number of slices modify ***ModelName\_input\_sij\_p2r.dat*** file in the CAVE/sara-sandbox/testing-b3pond /ss/ directory to add or remove lines as you see fit.
6. Open the ***[ModelName]-e-######.lay*** file.
7. In order to have the Python script access Tecplot click *Scripting 🡪 PyTecplot Connections* and then check the *Accept Connections* box.
8. Click onto the folder with the ***\*.bat*** file. Deselect all items, then hold Shift, right-click, and then click on “Open Command Window Here.”
9. Type ***run\_tecplotauto\_rotate\_slice.bat*** script and enter to run the script.
10. This script generates \*.png files in the /CAVE/sara-sandbox/testing-b3pond/ss/plts/figures directory.