



Tornado – Well tie Map

May 2016
Created By: D.Doledec

Passion for Geoscience



1. Open the Well Mistie Dialog and set the parameters for computation
 - Tornado > Horizons > analysis Tools > Well Misite Map
2. Click the Horizon annotation window
 - Tornado > Horizons > analysis Tools > Horizon Point annotations
3. Load the mistie text file by hand
4. repeat the setting for the display (tornado re-set everything into default setting when we load new file)
5. Grap the display picture from main window
6. Paste in the ppt



- In a python script
 1. Run the process well mistie map
 2. Load the misite text file & display
 1. Possibility to use display parameters (saved before form the Horizon Point annotations)
 3. Capture images



Loading data into the script

- If they are already loaded in Tornado

```
hg = vision.getCurrentHorizonGroup()
```

```
velocity = vision.getCurrentAttribute() # velocity for checkshots
```

- OR Load from file

```
hg_path='/data2/devtest/tornado/welltest/data/hrz_gb.hrz'
```

```
velocity_path='/data2/devtest/tornado/welltest/data/xp_M_a.fdm'
```

```
hg = HorizonGroup()
```

```
hg.load( hg_path )
```

```
velocity = Volume()
```

```
velocity.load(velocity_path)
```



Exemple1: compute misties for each couple (top/hrz)

- `processMisite = HorizonMisite()`
- `# Set mistie ascii File`
- `processMisite.setAnnotationFile(miscOutputFile)`
- `# Process`
- `topHorizonPair = [['Top_Balder','Top_Balder'],`
`['Top_Chalk','Top_Rattray'],`
`['Base_Rattray','Base_Rattray']]`
- `processMisite.run(hg, topHorizonPair)`



Exemple2: from a list of hrz (velocity model required)

- `processMisite = HorizonMisite()`
- `# Set mistie ascii File`
- `processMisite.setAnnotationFile(miscOutputFile)`
- `# Process`
- `horizons = ['Top_Balder','Top_Chalk','Herring']`
- `processMisite.run(hg, horizons, velocity)`



Parameters (exmple of parameters that can be changed)

7

Horizontal Interpolation: Triangulation, NatNeighbor, MinCurvature, Delta, Nearest

default: processMisite.setInterpolationMode(InterpMode.MINIMUM_CURVATURE, 0.2, 0)

example: processMisite.setInterpolationMode(InterpMode.LINEAR)

Vertical Interpolation for Missing tops (interpolation , output interpolated misite)

default: processMisite.interpolateMissingTops(False)

example: processMisite.interpolateMissingTops(True, True)

#Structure Based

default: processMisite.structureBased(False)

#output attribute name

default: processMisite.setOutputAttrName('MISTIE')

z in time Time or depth

default: 'DEPTH' if attr exist or 'TWT'

default: processMisite.setDomain('DEPTH')



Display the mistie ascii-file

Load Annotations created by the process

`annotations = PointAnnotation()`

Settings saved from the dialog

`annotations.loadDisplaySettings('/tmp/settingsAnnotations.xml')`

Set fields to display. As defined in the dialog: name, wellID, topName, hrzName, pos, depth, mistie, mistie_perc

Default: 'depth','mistie','mistie_perc'

`annotations.fields(['depth','mistie','mistie_perc'])`

Display misite extracted along the current hrz

`vision.setCurrentHorizon(hName)`

`annotations.load(miscOutputFile,1)`



Let's do something more complex ...

- Can be configured to work to load data or get data from the current session
- Can change the method of computation
 - Checkshot
 - Well top/hrz
- Load the information into Tornado (display)
- Loop through some horizons to display the QC
 - Misties for the displayed horizon

The python script is provided in

<http://red-qed/documentation/scripting/examples.html>

