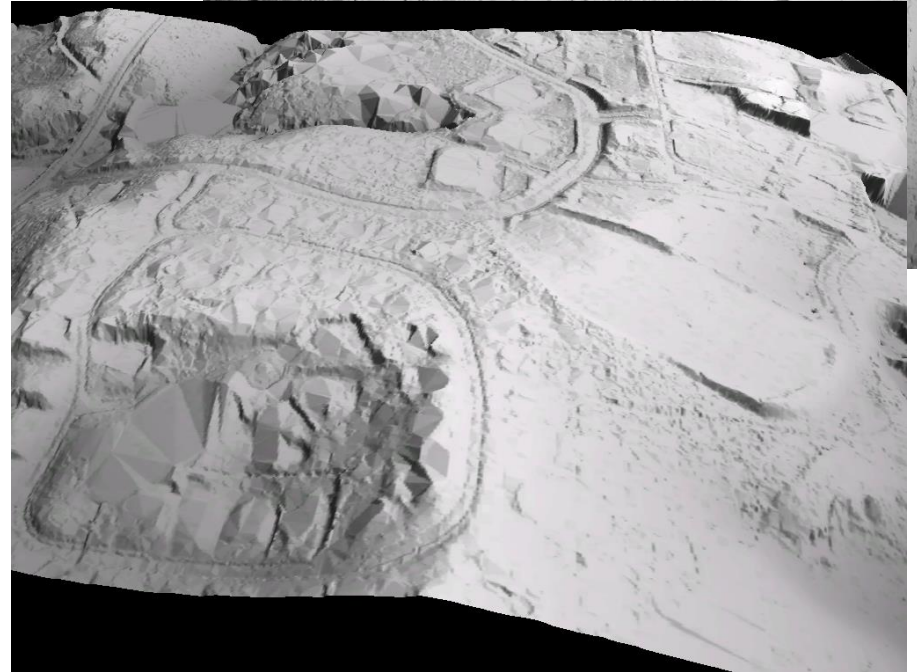
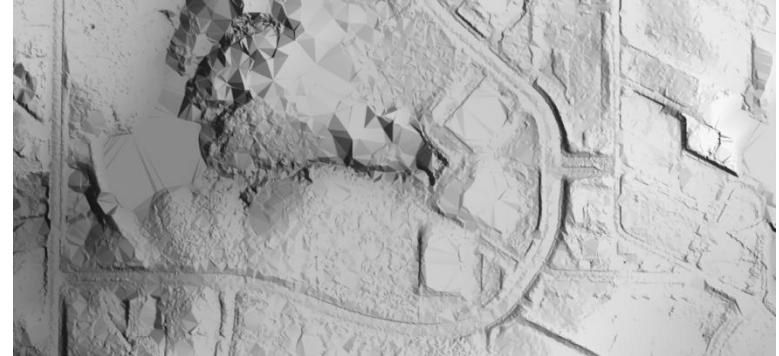


Managing PLS-CADD in the ESRI Environment

Matt Clarkson
Alabama Power

LiDAR / PLS-CADD

- What is LiDAR?
- PLS-CADD Deliverables
 - PLS Model
 - TIN
 - Ground/Veg. Points
 - LAS



The Problem



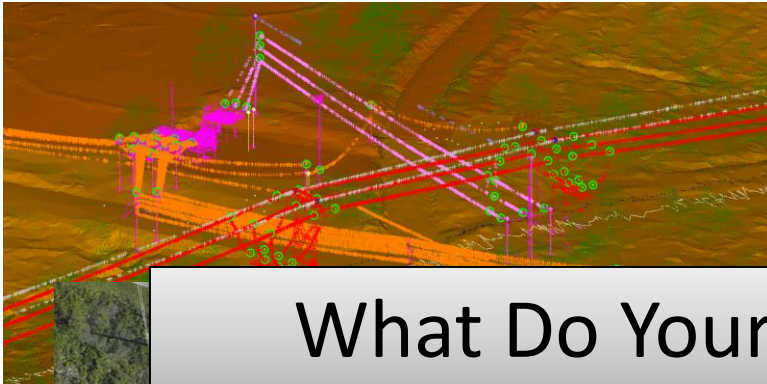
- NERC Facilities Alert / FAC-008
- Greenfield, Line Ratings, Other Projects
- Large Volume of Data
 - TBs of geospatial information
- Data Management

The Value

- Enterprise Usage
 - Construction
 - Environmental
 - Civil
 - Maintenance
- One-off Projects
 - 3D Visualizations
 - Line of Sight Analysis



APC's Solution



What Do Your Users Actually Need?

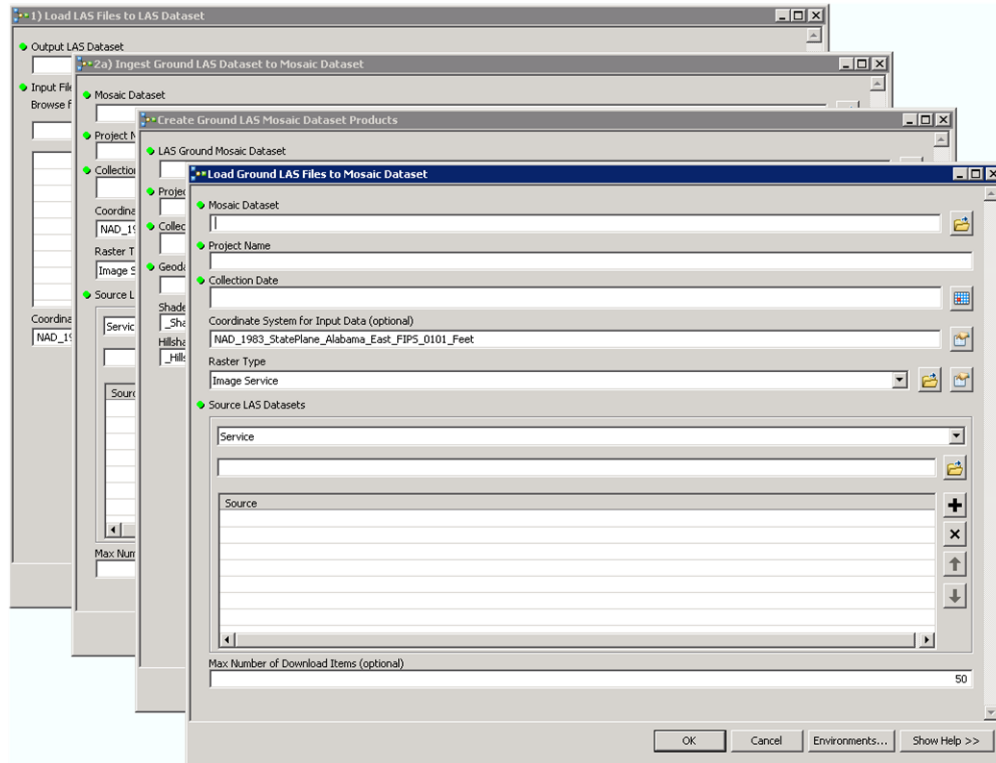
- Multiple Data Type
 - Match the data to the need
- Raster

vector

- Contours
- Point / Multipoint
- LAS
 - Data Management / Storage



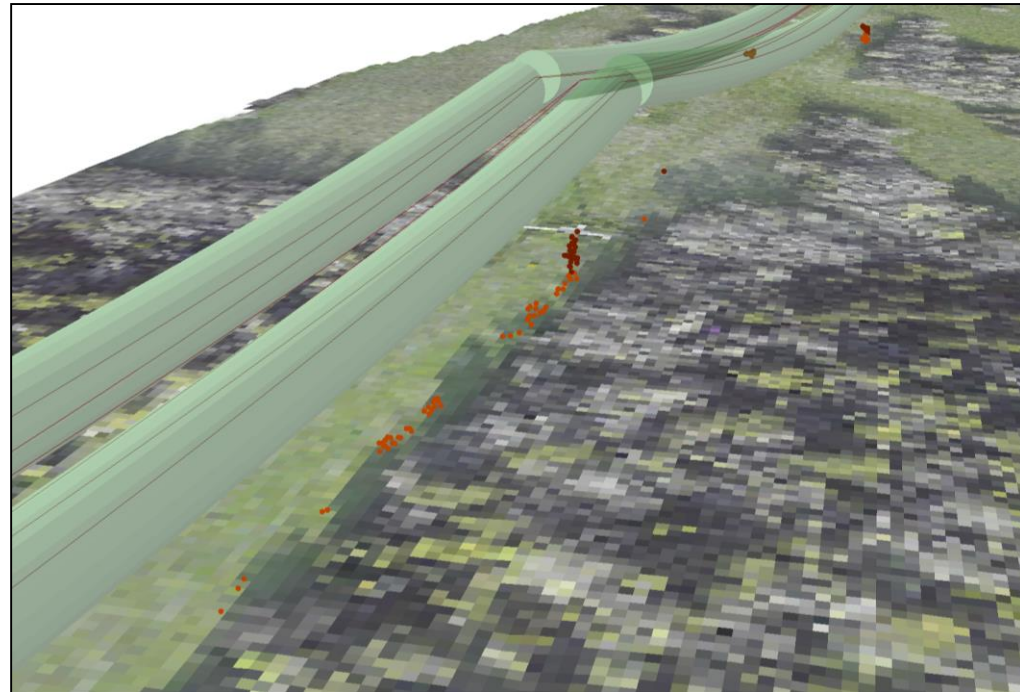
APC's Solution



- ESRI Image Server
 - Script Driven
 - Image and Vector Processing
- Conversion of PLS
 - Investment made in PLS Model Data
 - RGB Values Added
 - Small Width

APC's Solution

- Converting to Point
 - Python Script (In-line Processing)
 - ASCII PLS
 - Size considerations
- Domains
 - Feature Codes
- Additional Data
 - Alignment (NUM file)
 - Parser to Line Feature



APC's Solution

- Forth-Coming Improvements
 - LAS 1.4 Support
 - One to One Feature Code
 - Server solutions
 - Extractors
 - Web-Based

```
# Convert PLS to Shapfile
import arcpy
from arcpy import env
import fileinput
import string
import os

env.overwriteOutput = True

#Input XYZ in ASCII format
infile = arcpy.GetParameterAsText(0)
#Output file name
fname = arcpy.GetParameterAsText(1)

#Get Desired SPZ
SPZ = arcpy.GetParameterAsText(2)

#check for the selected SPZ
if str(SPZ) == "false":
    #Template (based on SPZ of the actual data)
    template = "<template path>\XYZ_Template_ALE.shp"
    spatRef = "<template path>\XYZ_Template_ALE.prj"
else:
    #Template (based on SPZ of the actual data)
    template = "<template path>\XYZ_Template_ALW.shp"
    spatRef = "<template path>\XYZ_Template_ALW.prj"

#Create shell of feature class
arcpy.CreateFeatureclass_management(os.path.dirname(fname), os.path.basename(fname), "POINT", template, "DISABLED", "ENABLED",
    spatRef)

#Create insert cursor
Inscur = arcpy.InsertCursor(fname, ("Nam", "SHAPE@X", "SHAPE@Y", "Z", "Fc", "Ht", "PrCom", "PlCom"))

ID = 0
#Read XYZ file
for line in fileinput.input(infile):
    #check to skip first line
    if ID == 0:
        ID = ID + 1
    else:
        pnt = arcpy.Point()
        pnt.Nam, pnt.X, pnt.Y, pnt.Z, pnt.Fc, pnt.Ht, pnt.PrCom, pnt.PlCom = string.split(line, " ")
        Inscur = Inscur.newRow()
        Inscur.set("SHAPE", pnt)
        Inscur.set("Nam", pnt.Nam)
        Inscur.set("X", pnt.X)
        Inscur.set("Y", pnt.Y)
        Inscur.set("Z", pnt.Z)
        Inscur.set("Fc", pnt.Fc)
        Inscur.set("Ht", pnt.Ht)
        Inscur.set("PrCom", pnt.PrCom)
        Inscur.set("PlCom", pnt.PlCom)
        Inscur.insertRow(Inscur)

        ID = ID + 1
fileinput.close()
del Inscur
```


The Wrap-Up

- Untapped Data = Untapped Value
- Selecting a Solution that Fits Your Data AND User Needs
- Empower Your Users by Allowing Access When You Can

Questions?