

```

# This python script will batch load very large .dat file exports (space delim)
# and create a GIS-friendly CSV files ready to import, that has also been down-sampled.
# Once the CSV is created it will then create a GIS XY feature layer and finally
# point feature class ready for interpolation.

# Chelsea Welker
# GEOG 6180
# December 2015

# import system modules
import arcpy
from arcpy import env
import os
from itertools import islice
import csv
import glob

# global settings
path = "C:/MSGIS/6180/___final/submit"
gdb_name = "test.gdb"

env.workspace = path
os.chdir(path) # changes directory to env.workspace

def convert_csv(txt_in, csv_out, num_strip = 0):
    """
    convert space delim text or dat file into comma delim csv file
    removing num_strip lines to down-size the input file if too large
    """

    if (os.path.exists(csv_out)) : # if the file is there (from previous run)
        os.remove(csv_out)         # then remove it

    in_txt = csv.reader(open(txt_in, "rb"), delimiter = ' ')
    out_csv = csv.writer(open(csv_out, 'wb'),)
    out_csv.writerow(["X", "Y", "Z"])

    if num_strip > 0:
        # this keeps one line and drops num_strip lines, and repeats
        # for large regional areas in Arc, and the very large text files, you can delete
        # between 10-30 rows.
        out_csv.writerows(islice(in_txt, 0, None, num_strip))
    else:
        out_csv.writerows(in_txt)

def csv_to_lyr(in_table, out_event_lyr, out_gdb, spatial_ref = 3395):
    """

```

converts the X,Y,Z csv into an ArcGIS feature class

Arguments:

in_table is a properly formatted csv

out_event_lyr is the ArcGIS XY Event Layer

out_gdb is where the new FC gets created

spatial_ref is the spatial given by the EPSG code or WKID

(<https://epsg.io/3395>)

WGS_1984_World_Mercator

WKID: 3395 Authority: EPSG

Geographic Coordinate System: GCS_WGS_1984

""

```
temp_lyr = r"._templyr.lyr"
```

```
out_name, ext = os.path.splitext(os.path.basename(in_table))
```

```
# Set the spatial reference
```

```
sr = arcpy.SpatialReference(spatial_ref)
```

```
# Make the XY event layer...
```

```
arcpy.MakeXYEventLayer_management(in_table, "X", "Y", out_event_lyr, sr, "Z")
```

```
# Save to a layer file
```

```
arcpy.SaveToLayerFile_management(out_event_lyr, temp_lyr)
```

```
#table to feature class
```

```
arcpy.TableToGeodatabase_conversion(temp_lyr, out_gdb)
```

```
arcpy.FeatureClassToFeatureClass_conversion(temp_lyr, out_gdb, out_name + "_fc")
```

```
if (os.path.exists(temp_lyr)) :
```

```
    os.remove(temp_lyr)
```

```
# main
```

```
files = glob.glob('./*.txt') # finds all text files in the current directory
```

```
#arcpy.CreateFileGDB_management('./', gdb_name)
```

```
# runs the script for all .txt files in the directory
```

```
for f in files:
```

```
    name, ext = os.path.splitext(os.path.basename(f))
```

```
    convert_csv(name + ext, name + '.csv', 10)
```

```
    csv_to_lyr(name + '.csv', name, gdb_name)
```