Precision Stalker Script – Background

After a few different iterations of trying to provide vector data overlays (KML) for both the Stalker GCS and Skylink gimbal software, we found the most reliable way to get a consistent product was to create GeoTiffs with the vector data baked in over the preferred background (imagery for GCS, topo for gimbal). Initially, we started with high DPI (1200 dpi) single GeoTiff images, but we found that at some point increasing the dpi doesn't result in a more detailed export image. Due the size of the fire/flight-area and the desired resolution of the background imagery for safety of flight, we switched to a Grid-Index-style tiling approach.

GlobalMapper is capable of easily grid tiling an existing map/image and exporting it as a GeoTiff with tags, but ArcPro's Map Series function does not provide out-of-the-box export to a GeoTiff with tags. With ArcPro, one can create a Grid Index, manually zoom to each grid, and export each manually, but that gets cumbersome beyond a certain number of tiles (20? 30?). To avoid the hassle of having to work in both GlobalMapper and ArcPro and reduce the potential error involved in a large number of manual exports for each GCS product, I created a workflow/script to aid the process.

Workflow & Scripts

There's a couple baseline assumptions with the script:

1. You need to create a grid index feature class in WGS84 to feed the script. (I built mine to be 8"x 8" pages at 1:24,000). You'll need to have that grid index feature added to a Map tab with the coordinate system set to WGS84.
2. You need to have a Map Layout that matches the size of your Grid Index features with the Map created in step 1 as the referenced data frame. You'll need the names of both the Map and the Layout elements to feed the script.

There are two snippets of code that comprise the script. Each are run from the Python window inside ArcPro. The first part iterates through the features in your grid index feature class and creates a bookmark file based on the extent of each feature. You then import the bookmark file into your project and execute the second snippet. The second snippet loops through the bookmarks and exports each to a GeoTiff with tags.

Snippet #1

```
# coding: utf-8

##Bookmark generation RUN ME FIRST

aprx = arcpy.mp.ArcGISProject("CURRENT")

map = aprx.listMaps("GCS MAP")[0]

lyt = aprx.listLayouts("GCS_Tiles")[0]

#this is for WGS84

out = """{

  "type" : "CIMBookmarkDocument",
  "version" : "2.5.0",
  "build" : 22081,
  "spatialReference" : {
    "wkid" : 4326,
    "latestWkid" : 4326
  },
  "bookmarks" : [
"""
LE = []
with arcpy.da.SearchCursor("Riverside_GridIndex_20200918", "SHAPE@") as cursor:
    for row in cursor:
        ext = row[0].extent
        thisL = [ext.XMin, ext.YMin, ext.XMax, ext.YMax]
        LE.append(thisL)
x = 0
for ext in LE:
    out += '{\n"type" : "CIMBookmark",\n"name" : "Bookmark %s",\n"camera" : {\n},\n"location" : {\n' % str(x)
    out += '"xmin" : %s,\n' % str(ext[0])
    out += '"ymin" : %s,\n' % str(ext[1])
    out += '"xmax"  : %s,\n' % str(ext[2])
    out += '"ymax" : %s,\n' % str(ext[3])
    out += '"spatialReference" : {\n'
    out += '"wkid" : 4326,\n'
    out += '"latestWkid" : 4326\n'
    out += "}\n"
    out += "}\n"
    out += "},\n"
    x += 1
out = out [:-2] + "\n]\n}\n"
f = open(r"D:\2020_Fires\2020_UASD_Riverside_ORMHF_000859\UAS_Data\Riverside_bookmarks.bkmx", 'w')
f.write(out)
f.close()
```

The highlighted elements are parameters that you have to change to match your Map name, Layout name, Grid Index name, and the location/name of your bookmark file.

Once you have a bookmark file generated, you’ll need to remove any existing Bookmarks you have associated with the Map you are using as the script will iterate through all bookmarks indiscriminately. You can remove bookmarks by going to Map > Bookmarks > Manage Bookmarks and clicking the red X next to each bookmark. You can then import your new Bookmark file by hitting the burger button and going to Import Bookmarks. With the bookmarks imported, you can run the next snippet.

Snippet #2

```
##Tiles Export RUN ME SECOND AFTER IMPORTING BOOKMARKS
aprx = arcpy.mp.ArcGISProject('current')
map = aprx.listMaps("GCS_Map")[0]
lyr = map.listLayers("BeachieCreek_GridIndex_20200918")[0]
lyt = aprx.listLayouts("GCS_Tiles")[0]
mf = lyt.listElements('MAPFRAME_ELEMENT',"Map Frame")[0]
books = map.listBookmarks()
x=1


for book in books:
   mf.zoomToBookmark(book)


mf.exportToTIFF(r"D:\2020_Fires\2020_UASD_Beachie_Creek_ORWIF_200299\UASD_Final_Products\GCS_Maps\20200921\BeachieCreek_GCS_Tile_" + str(x) + ".tif",400,False, '24-BIT_TRUE_COLOR', 'LZW',True)
   x += 1
```

Again, the highlight elements are parameters you will have to change to match your Map name, Grid Index name, Layout name, Export path & filename, image resolution (dpi), and bit-depth