ArcGIS for Imagery

Presented by:
Brendan O’Neill, Solution Engineer

GIS FOR A SUSTAINABLE WORLD 2018
Thank You to Our Sponsors
ArcGIS Imagery Content

- World Imagery
- World Landsat
- World Elevation
- NAIP
- HYCOM
- Sentinel 2
The ArcGIS Platform
Is a Comprehensive Imagery Platform

System of Insight
Extract Information

Professional Imagery & Geospatial Analysts

Enterprise

Content from: ArcGIS Online, Partners, Your Org.

System of Record
Manage and process your imagery

System of Engagement
Share imagery products and information
Mosaic Datasets are the Cornerstone

Imagery Services
OGC & KML

Multiple Sensors & Formats

Content Management
- Shared Data Model
- Data Discovery
- Metadata Handling
- Time Enabled

Mosaic Datasets:
An Imagery Information Model

Web Cataloging - Search & Discovery
Share and Discover Image Content

- Query
- Return Results
- Preview
- Use
Mosaic Dataset Configuration Script

- A single Python tool to create, populate and configure a mosaic dataset
- Wraps all necessary GP tools into one
- Input = 1 configuration file
- GPTool or Command line or ModelBuilder
- Many Work arounds - Handles different versions of ArcGIS

`python.exe mdcs.py -i:configfile.xml`

https://github.com/Esri/mdcs-py
MDSCS Command Codes

- CM = Create Mosaic Datasets
- AF = Add Fields
- AR = Add Rasters
- CR = Create Referenced MDs
- BF = Build Footprints
- BB = Build Boundary
- IF = Import Fields
- IG = Import Geometry
- DN = Define NoData values
- CBMD = Color Balance MD
- ERF = Edit Raster Function
- SP = Set properties
- SS = Set statistics
- CC = Calculate cell size ranges
- BO = Build overviews
- DO = Define overviews
- AI = Add index
- CV = Calculate values
- BP = Build Pyramid
- JF = Join Fields
- BPS = Build Pyramids and Statistics
- …..
Mosaic Dataset Configuration Script Workflow
Demo
Sharing Imagery as Web Layers

- To ArcGIS Enterprise
  - Map Image Layer
  - Image Layer
  - Tile Layer
- To ArcGIS Online
  - Tile Layer

<table>
<thead>
<tr>
<th></th>
<th>Bands</th>
<th>Analysis</th>
<th>Speed</th>
<th>Metadata</th>
<th>Query</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Map Image Layer</strong></td>
<td>3-band, 8-bit</td>
<td>No</td>
<td>Highly Scalable</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Tile Layer</strong></td>
<td>3-band, 8-bit</td>
<td>No</td>
<td>Highly Scalable</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Image Layer</strong></td>
<td>Multiple bands, dimensions, bit depths</td>
<td>Yes</td>
<td>Dependent on resources</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Sharing Imagery as an Image Layer
Enterprise GIS setup
Sharing Imagery as an Image Layer (Reference)

Publishing mode

- **Share by reference**
  - Register a file share and/or enterprise geodatabase(s) with the server data store
  - Data is not moved
  - Recommended for publishing mosaic dataset or large single images or rasters
Running ArcGIS Image Server in the Cloud

- **Advantages**
  - Lower cost resilient storage
  - Lower cost enterprise compute
  - Simpler install
  - Simpler scalability

- **Disadvantages**
  - Data needs to be uploaded
  - Different storage types
  - Infrastructure changes
  - Potential security concerns
  - Potential for complex data access policies
Deployment Tools - AWS Cloud Formation templates

- Need base ArcGIS Enterprise deployment
- Multi-server site must have ELB
- ArcGIS Server install (Image Server role)

<table>
<thead>
<tr>
<th>Template Name</th>
<th>Description</th>
<th>View</th>
<th>Launch</th>
</tr>
</thead>
<tbody>
<tr>
<td>ELB for ArcGIS GIS Server deployment</td>
<td>Create an Elastic LoadBalancer that can be used for ArcGIS GIS Server deployment.</td>
<td></td>
<td>LAUNCH STACK</td>
</tr>
<tr>
<td>ArcGIS GIS Server</td>
<td>This template creates a general purpose GIS Server site, or GeoAnalytics Server site, or RasterAnalytics Server site that you can federate with the portal in your base ArcGIS Enterprise deployment. You just need to use different Server license file to deploy different Server sites. ReadMe</td>
<td></td>
<td>LAUNCH STACK</td>
</tr>
</tbody>
</table>

https://s3.amazonaws.com/arcgisstore106/8321/docs/index.html
Deployment Tools - Cloud Builder Command Line Interface for AWS

- New at 10.6
- One Run to Deploy ArcGIS Enterprise
  - Base + Federated Servers
  - Standalone server
  - Define in a JSON file
- Prepare your deployment
  - Create VPC
  - Create ELB
  - Upload license and install certificate

Deployment Tools - ArcGIS Cloud Builder for Azure


ArcGIS Enterprise in the cloud

Install ArcGIS Enterprise Cloud Builder for Microsoft Azure

ArcGIS Enterprise Cloud Builder for Microsoft Azure is an application you install on your local Windows machine to deploy ArcGIS Enterprise and stand-alone ArcGIS Server sites on Microsoft Azure. It helps you to extend your Azure implementation to include ArcGIS.

Make sure your local Windows machine meets the following prerequisites, download the installation file, and install.
Demo
Imagery Analysis - Components

ArcGIS Pro – Image Analyst Extension & Spatial Analyst Extension

ArcGIS Image Server
- Dynamic Image Services
- Raster Analytics
- Hydro Tools

System of Insight

Image Space & Mensuration
Classification
Change Analysis
Full Motion Video
Stereo Mapping
Raster Analysis Clients

**ArcGIS Pro**

**ArcGIS Enterprise**

**Python API**
System of Insight
Geoprocessing, Raster Function, Image Service

Extract Information from Imagery

Desktop
Geoprocessing

Server on-the-fly
Image Service Processing Template

Enterprise
Distributed Raster Analysis

Multi-server distribution
Raster Functions in Pro

- Out-of-the-box functions
- String together function chains
- Import custom function templates
Raster Function Chains
Raster Function Processing

- **Client side (ArcGIS Pro)**
  - Pixels are processed locally
  - Processing is done On-the-fly

- **Server side (ArcGIS Enterprise/Pro)**
  - Pixels are processed on the server
  - Processing is done On-the-fly

- **Enterprise (ArcGIS Enterprise)**
  - Distributed processing of large jobs on the server
  - Results are persisted
Enterprise Image Server with Distributed Raster Analysis

- Complete analysis task for data too big for single desktop machine
  - Meet time constraint (*Month* -> *Weeks* -> *Days* -> *Hours* -> *Minutes*)

Dynamic Raster Models

Geoprocessing Models

Server-based Distributed Raster Analytics with Distributed Raster Data Storage

(on persistent) distributed analytics with optional distributed storage for even greater scalability

Web GIS Layers

rich geoinformation model
Raster Analytics Test Case: Landsat Processing

Infrastructure
- Esri Web GIS on AWS
- Distributed Raster Analytics Cluster
  - single node
  - AWS c3.8xlarge
  - 60GB RAM, 32 cores, 500GB SSD
  - 200 Raster Analytics Processors

Input Collection
- Landsat GLS 1990
  - 7422 Multispectral Scenes
  - S3 storage

Processing
1. (foreach) input scene
2. mask no data
3. top of atmosphere correction
4. modified soil adjusted vegetation index
5. remap to classes
6. output thematic raster

Output
- Thematic Rasters
  - 7422 Thematic Rasters
  - Distributed Raster Datastore

Time:
- 2 hours 48 minutes
- 44 scenes per minute
- ¾ scene per second
ArcGIS Python API overview

- The ArcGIS API for Python is implemented using the online and on-premises web GIS platform provided by ArcGIS Online and ArcGIS Enterprise respectively.
- Distributed as the `arcgis` conda package.
- `arcgis.raster` module
  - The `arcgis.raster` module contains classes and `raster analysis functions` for working with raster data and imagery layers in ArcGIS Enterprise.
ArcGIS Python API for Raster Analysis - Image Layer

• Display Image Layer
• Apply Image Service on-the-fly processing
ArcGIS Python API for Raster Analysis – Raster Functions

- Construct custom raster function chain
- Preview analysis result
- Submit raster analysis task

```python
# Construct raster analysis workflow using raster function
# 1. Create slope and aspect raster from DEM
elev_slope = slope(elev, slope_type="DEGREE", z_factor=1)
elev_aspect = aspect(elev)

# 2. Apply suitability condition on slope raster (5 < slope < 60)
slope = boolean_and((greater_than(elev_slope, 5)), less_than(elev_slope, 60)))

# 3. Apply suitability condition on aspect raster (0 < aspect < 200)
saspect = boolean_and((greater_than(elev_aspect, 0)), less_than(elev_aspect, 200)))

# 4. Apply suitability condition on DEM (30 < DEM < 400)
selev = boolean_and((greater_than(elev, 30)), less_than(elev, 400)))

# 5. Sum up using map algebra
output_values = [0, 64, 128, 255]
result = colormap(cmap(slope +aspect + selev,
input_ranges = [0,0,1,2,3,3],
output_values = output_values),
colormap="Red To Green")
```
Demo
Engage with Imagery - Tools
ArcGIS Applications Enable Intelligent Imagery

- Dynamic & Interactive
- Informative & Engaging
- Integrated

WABIS
Web App Builder for Image Services

Imagery Templates

GitHub Samples

Java Script APIs & Web Map Viewer