ArcGIS Workflows for Optimizing Image Management & Services in the Cloud

Cody A. Benkelman
Outline

- Mosaic Datasets
- Automation with the Mosaic Dataset Configuration Script (MDCS)
- OptimizeRasters

*Lower level & more detailed discussion in*
- Automating Image Management and Dissemination using Python
- Friday, March 10 8:30 AM – 9:30 AM  Santa Rosa
Mosaic Dataset

Optimum Model for Image Data Management

- Catalog all types of raster data
- Image data remains in original files
- Define – in Geodatabase:
  - Metadata
  - Processing to be applied
  - Default viewing rules
- Access – In all ArcGIS applications
  - As Image
    - Dynamic Mosaic, Processed on the fly
  - As Catalog
    - Footprints, Detailed metadata
Mosaic datasets: powerful, flexible... with many details

- MD properties
  - Configuration depends on data type and also use case
Mosaic datasets: powerful, flexible... with many details

- MD properties
  - Configuration depends on data type and also use case
- GP Tools
  - Many have multiple options
Demo – Landsat 8

8 band image service
On-the-fly raster functions
300 GB updated daily, 100% automated
Automation with MDCS

• Productivity
  - Repeatability, Scalability, Maintainability
  - Documentation → Facilitate QA & QC, Design Review

• Training/Examples
  - Encapsulate best practices
  - Reusable templates

• Simplicity
Options for building Mosaic Datasets

- Discrete GP tools (manual)
- ModelBuilder
- Python (since 10.1)
Create Simple Mosaic Dataset - ModelBuilder
Comparing Models
Python in the GP Framework

- Calling Geoprocessing tools in Python:

  ```python
  Import arcpy
  arcpy.CreateMosaicDataset_management(“C:\temp\test.gdb”, ”abc”…
  ```

- HELP LINK  Using tools in Python

- HELP LINK  Create Mosaic Dataset

Automating Image Management and Dissemination using Python
- Friday, March 10  8:30 AM – 9:30 AM  Santa Rosa
Python implementation - *Mosaic Dataset Configuration Script (MDCS)*

- Calling standard Geoprocessing tools from a single script
- Input configuration file contains complete information to:
  - Create,
  - Populate, and
  - Configure one mosaic dataset

- Also generates detailed log files
Advantages of MDCS

- Configuration file encapsulates “Best practices” (mosaic dataset properties) based on image type

- “Self Documenting” –
  - Template is reusable for different image types, or multiple mosaic datasets within a more complex system
  - Compare versions (difficult with ModelBuilder)

- Automated Log files – Simple Review

- Extensible: additional commands can be added
Other features to note within MDCS

- Can run subsets of full configuration via command line options
- Built in version compatibility checks
- Extensible: additional commands can be added
Configuration file contents

Input Data Paths
Configuration file contents

- Input Data
- Paths
- Raster Types
- & Raster Functions
Configuration file contents

- **Input Data Paths**
- **GP tools necessary for the workflow**
- **Raster Types & Raster Functions**
- **Mosaic Dataset properties**

[Diagram showing the relationships between different components of the configuration file contents.]
Configuration file contents

- Input Data Paths
- GP tools necessary for the workflow
- Mosaic Dataset properties
- Raster Types & Raster Functions

“BEST PRACTICES”
Configuration XML file

```xml
<AddRasters>
  <DefaultProperties>
    <RasterPartMosaic>50</RasterPartMosaic>
    <MaxRequestSizeX>4000</MaxRequestSizeX>
    <MaxRequestSizeY>4000</MaxRequestSizeY>
    <allowed_compressions>LZ77;NONE;LERC</allowed_compressions>
    <default_compression_type>LERC</default_compression_type>
    <CompressionQuality>75</CompressionQuality>
    <resampling_type>BILINEAR</resampling_type>
    <LERC_Tolerance>0.01</LERC_Tolerance>
    <clip_to_footprints>CLIP</clip_to_footprints>
    <clip_to_boundary>CLIP</clip_to_boundary>
    <color_correction>NOT APPLY</color_correction>
    <footprints_may_contain_nodata>FOOTPRINTS_MAY_CONTAIN_NODATA</footprints_may_contain_nodata>
    <allowed_mensuration_capabilities>BASIC</allowed_mensuration_capabilities>
    <default_mensuration_capabilities>BASIC</default_mensuration_capabilities>
    <allowed_mosaic_methods>LockRaster;ByAttribute;Seamline;None</allowed_mosaic_methods>
    <default_mosaic_method>ByAttribute</default_mosaic_method>
  </DefaultProperties>
  <Order_field>BEST</Order_field>
  <order_base>0</order_base>
  <sorting_order>Ascending</sorting_order>
  <mosaic_operator>FIRST</mosaic_operator>
  <blend_width>10</blend_width>
  <view_point_x>300</view_point_x>
  <view_point_y>300</view_point_y>
  <max_num_per_mosaic>50</max_num_per_mosaic>
  <cell_size_tolerance>999</cell_size_tolerance>
  <cell_size>0</cell_size>
  <metadata_level>BASIC</metadata_level>
  <transmission_fields>Name;MinPS;MaxPS;LowPS;HighPS;ProductName;BEST;Source;LE90;CE90;D
  <use_time>DISABLED</use_time>
```
MDCS via GP interface (Landsat 8 example)

- Geoprocessing tool interface for end users.
- Allows users to define output GDB, mosaic dataset name and input directory.
- Checkboxes are used to trigger elements of MDCS that create products.
- Allows users to output mosaic dataset as well as referenced mosaic dataset products.
- Customizable
MDCS extensibility: User Code module

```python
MDCS_UC.py

class UserCode:
    def A(self, data):
        workspace = data['workspace']
        md = data['mosaicdataset']
        log = data['log']
        log.Message('%s\n%s\n' % (workspace, md), 0)
        return True

    def B(self, data):
        log = data['log']
        log.Message('hello world', 0)
        return True

User function (A)
User function (B)

C:\> python.exe MDCS.py -c:CM+A+B+AR
```
Demo - MDCS

Preprocessed Orthophotos
Managing large image collections

- Best practices implicit in the design of MDCS
- Key field in configuration file “MD Type”, with allowable values:
  - Source
  - Derived
  - Referenced

For simple systems, use “Source” and stop here!
Source Mosaic Datasets

Source Imagery → Source Mosaic Datasets

SRTM

NED 1/3 arcsec

LiDAR Project #N
Source Mosaic Datasets

Source Imagery → Source Mosaic Datasets

- 2004
- 2007
- 2010
Combine into Derived Mosaic Dataset

Source Imagery -> Source Mosaic Datasets -> Derived Mosaic Dataset

Advantage: All image data available in a single location
Example – ArcGIS World Elevation – Server Raster Functions

Source Imagery

Source Mosaic Datasets

Derived Mosaic Dataset

Single image service with multiple server functions

Multispectral image service

RGB
CIR
NDVI
Pan sharpen
...many other functions

Preferred architecture added at 10.1
Example – ArcGIS World Elevation

Source Imagery → Source Mosaic Datasets → Derived Mosaic Dataset → Single image service with multiple server functions

- Multispectral image service
  - RGB
  - CIR
  - NDVI
  - Pan sharpen
  - ...many other functions

When Derived parent is updated, Referenced Mosaic Datasets synch automatically
Finding MDCS and related resources
http://esriurl.com/ImageryWorkflows

• Landing page on Resource Center

• ArcGIS Online (AGOL) group with downloadable tools
OptimizeRasters
GitHub: http://esriurl.com/OptimizeRasters

- Copy to Cloud storage (AWS, Azure)
- Optimize format
  - Internal tiling & pyramids
  - Optional MRF format & LERC compression

http://esriurl.com/MRF
NASA format, Open Source
Used for Esri’s World Elevation Service & NAIP image service
Resources

• Landing page: http://esriurl.com/ImageryWorkflows
• Guidebook in ArcGIS Help: http://esriurl.com/6007
• ArcGIS Online Group: http://esriurl.com/6539
• MDCS Available on Github: http://esriurl.com/MDCS
• Optimize Rasters: http://esriurl.com/OptimizeRasters
• MRF format: http://esriurl.com/MRF

- Cody Benkelman  cbenkelman@esri.com