Automating Workflows for Creating and Publishing Imagery Using Mosaic Datasets

Cody Benkelman & Jamie Drisdelle

Wednesday, March 27, 2013  4:00 pm
SESSION ID: 90
OFFERING ID: 269
Outline

- Mosaic Datasets
- Automation with ModelBuilder & Python
- *Mosaic Dataset Configuration Script*
  - Configuration files
  - Source / Master / Derived design
  - Documentation and other resources
Mosaic datasets: Content & Creation
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

- GP Tools
  - Many have multiple options
Mosaic datasets: powerful, flexible… with many details

- **GP Tools**
  - Many have multiple options
- **MD properties**
  - Configuration depends on data type and also use case
Example Workflow

Create Mosaic Dataset

Add Rasters & Set Raster Type Properties

Set Properties

Define NoData

Add Fields & Calculate Values

Calculate or Set Statistics

Calculate Cell Size Ranges

Build Boundary

Add Original Mosaic Datasets to Derived Mosaic Datasets

Create Referenced Mosaic Datasets

Create Derived Mosaic Dataset

Add Functions
Example Workflow

Create Mosaic Dataset
Add Rasters & Set Raster Type Properties
Set Properties
Define NoData
Add Fields & Calculate Values
Calculate or Set Statistics
Calculate Cell Size Ranges
Build Boundary
Create Derived Mosaic Dataset
Add Original Mosaic Datasets to Derived Mosaic Datasets
Create Referenced Mosaic Datasets
Add Functions
Automation Requirements

• Simplicity
• Efficiency
  - Repeatability & Scalability
  - Documentation ‡ Facilitate QA & QC, Design Review
• Training/Examples
  - Encapsulate best practices
  - Reusable templates
Options for building Mosaic Datasets

• Discrete GP tools (manual)
• ModelBuilder
• Python (added at 10.1)
Python in the GP Framework

• Calling Gptools in Python:
  - Import arcpy
  - `arcpy.CreateMosaicDataset_management("C:\temp\test.gdb", "abc"...)`
  - HELP LINK Using tools in Python
  - HELP LINK Create Mosaic Dataset

• Creating tools using Python tool box:
  - an integral part of geoprocessing
  - use in ModelBuilder, and call it from scripts
  - HELP LINK What is a Python toolbox?
  - HELP LINK Creating a new Python toolbox
Python Example - *Mosaic Dataset Configuration Script*

- A compilation of standard GP tools into one script
- Input configuration file contains complete information to:
  - Create,
  - Populate, and
  - Configure one mosaic dataset

- Also generates detailed log files
Simple end to end tool for Landsat
Simple end to end tool for Elevation
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
- Can run subsets via command line options
- Open Source – Github – Modify as required
Configuration file contents

Input
Data
Paths
Configuration file contents

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

- Input
- Data
- Paths
- Raster Types & Raster Functions
- Mosaic Dataset properties
Configuration file contents

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

Input Data Paths

GP tools necessary for the workflow

Mosaic Dataset properties

"BEST PRACTICES"

Raster Types & Raster Functions
Setting configuration file parameters

CFE (configuration file editor)
<DefaultProperties>
  <RasterPerMosaic>50</RasterPerMosaic>
  <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>
  <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>#</cell_size>
  <metadata_level>BASIC</metadata_level>
  <transmission_fields>Name;MinPS;MaxPS;LowPS;HighPS;ProductName;BEST;Source;LE90;CE90;D
Correlating XML to MD Properties

```
<Filter>*</Filter>
/AddRaster
/AddRasters
/defaultProperties>
<RasterPerMosaic>50</RasterPerMosaic>
<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>
```

- **Image Properties**
  - Max Size of Requests:
    - Rows: 4000
    - Columns: 4000
  - Allowed Compression Methods: None, LERC
  - Default Compression Type: LERC
  - Compression Quality: 75
  - Resampling Type: BILINEAR
  - LERC Tolerance: 0.01
  - Always Clip the Raster to its Footprint: Yes
  - Always Clip the Mosaic Dataset to its Boundary: Yes

- **Catalog Properties**
  - Raster Metadata Level: Full
  - Maximum Number of Records Returned: 1000
ArcGIS Online (AGOL) group

- Downloadable examples ("elevation" available now)
- Multiple workflows/templates to be added over time
Configuration file UI for World Elevation
Multi dataset, multi resolution
Managing large image collections

- Next 4 slides come from another presentation…
- Best practices implicit in the design of MDCS
- Key field in configuration file “MD Type”, with allowable values:
  - Source
  - Derived
  - Referenced

  ![XML Code]

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

Source Imagery → Source Mosaic Datasets

“2004”

“2007”

“2010”
Source Mosaic Datasets

Source Imagery → Source Mosaic Datasets

“SRTM”

“NED 1/3 arcsec”

“LiDAR Project #N”
Combine into Derived Mosaic Dataset

Source Imagery → Source Mosaic Datasets → Derived Mosaic Dataset

Advantage: All image data available in a single location
Referenced Mosaic Datasets

- Read-only copies of the parent MD
- Functions applied on-the-fly; no need to write new raster to disk
- Footnote: At 10.1, functions can be applied on the server instead of separate Ref MDs

Limited extent e.g. "California"
- Pan sharpening
- NDVI
Example – ArcGIS World Elevation

When Derived parent is updated, Referenced Mosaic Datasets synch automatically
Simple workflow for publishing

• Get mosaic dataset
• Set data store settings on server for *publishing by reference*
• Create service definition draft
• Edit image service settings in service definition draft
• “Stage” the service definition draft to *.sd file
• Upload *.sd file to server

• For more information – see session #309 Thursday…
Resources

• Resources re: “Imagery Management Workflows”
  - Imagery Resource Center
    - Quick Link to Image Management Workflows
  - “Image Management Workflows” Guidebook in ArcGIS Help
  - GitHub: https://github.com/ArcGIS/mdcs-py
    - Must have an account and sign in

• Presenters:
  - Cody Benkelman cbenkelman@esri.com
  - Jamie Drisdelle jdrisdelle@esri.com

PLEASE FILL OUT A SURVEY! Offering ID 269
EXTRA SLIDES AFTER THIS POINT

Stop....
Batch File / Command Line Syntax

python.exe  mdcsl.py  –i:configfile.xml
<?xml version="1.0"?>
<Application>
  <Name>Elevation_Project</Name>
  <Command>CM+AF+AR+CC+CV+BB+SP+SS</Command>
  <Workspace>
    <WorkspacePath>MD</WorkspacePath>
    <Geodatabase>Sample2.gdb</Geodatabase>
    <MosaicDataset>
      <MosaicDatasetType>Source</MosaicDatasetType>
      <Name>S_SRTM</Name>
      <SRS>PROJCS['WGS_1984_web_Mercator_Auxiliary_Sphere',GEOGCS['WGS_1984',DATUM['D_WGS_1984'],SPHEROID['WGS_1984',6378137.0,298.257223563,GRD_Greenwich,0.0],UNIT['Degree',0.0174532925199433]],PROJCS['Mercator_Auxiliary_Sphere'],PARAMETER['False_Easting',0.0],[False_Northing',0.0],PARAMETER['Central_Meridian',0.0],PARAMETER['Standard_Parallel_1',0.0],PARAMETER['Auxiliary_Sphere_Type',0.0],PARAMETER['False_Easting',0.0],PARAMETER['False_Northing',0.0]]</SRS>
      <num_bands>1</num_bands>
      <PixelType>32_BIT_FLOAT</PixelType>
    </MosaicDataset>
  </Workspace>
</Application>
MDCS Details

- Single Entry point to a collection of Python Classes
- Checks for input parameters (If not present it uses values defined in the XML)
- Parses xml strings
- Calls logging and solutionlib modules
- Adds and populates “dataset id” fields.
SHORT 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

- 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
Other scripts

- Add fields
- Add rasters
- Create MD
- Create referenced MD
- Process Info
- Set MD properties
- Logger
Source / Derived / Referenced applied in MDCS
Source / Derived / Referenced applied in MDCS
“Scripts” download
“Scripts” download
“Scripts” download
Image Management Workflows

Manage collections of imagery in ArcGIS using a mosaic dataset.
Share your collections online as image services.

ArcGIS enables you to work with a wide variety of imagery acquired from different sources. These image management workflows provide best practices for managing large collections of imagery to make the imagery quickly and efficiently accessible.

The image management workflows include a general overview of all aspects related to image management and the following components for each workflow:

- Detailed review of best practices.
- Scripts and associated documentation for the automation of workflows.
- Sample data to be used with the scripts to create sample mosaic datasets and image services.
- Links to live example services on ArcGIS Online.
- Links to other resources such as Forums, Blogs, and Help system.
- A Image Management Guidebook that provides an overview of image management best practices.

Following are the workflows that are currently documented or in the process of being documented:

Elevation

This workflow describes organization and sharing of elevation data from different sources in multiple resolutions, and will need to be maintained over time as new data (e.g., from Lidar) is acquired. Derived visualizations such as hillshades may be generated on the fly.

Elevation or terrain data is used for many applications—in some cases to directly extract information, such as computing viewsheds, contours and profiles—and also forms the basis for creating detailed scene analytics and other complex visualizations.