

Esri Developer Summit

March 26–29, 2012 | Palm Springs, California

esri.com/events/devsummit



Programming Image Services using the Silverlight Web API

Wenxue Ju, Hong Xu

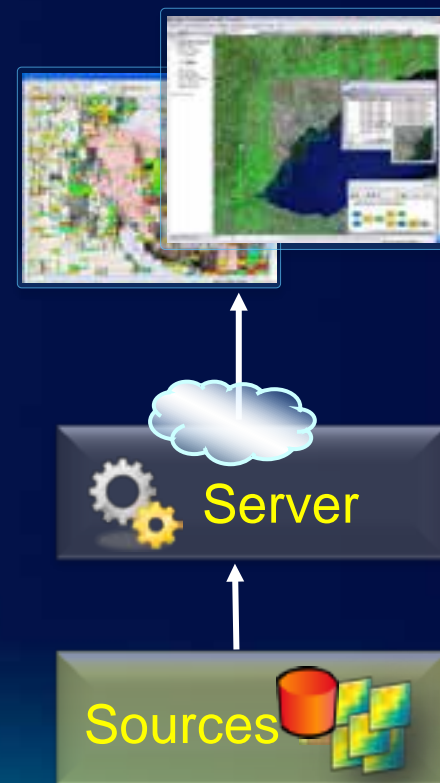


Presentation Outline

- **Introduction of image service**
- **Image service capabilities**
- **Developing image service web application**
 - **Discuss seven application patterns**

What is an Image Service?

- An ArcGIS Server service that provides image data and the associated functionality through web application
 - A single image
 - A collection of images as a mosaic
 - A collection of images as a catalog
 - Dynamic processing
- New in 10.1 : image service cache
- Image service can be dynamic and static



What Can Image Services Offer?

- **Fast display**
- **Export data**
 - Pixel value, raw image, and processed image
- **Mosaicking and cataloging**
- **Dynamic image processing**
- **Image mensuration**
- **Image service editing**
 - Add, delete, update
- **Support OGC standard**
 - WCS, WMS, WMTS, KML

Image Service Clients

- REST, SOAP
- Web APIs
 - Silverlight, Flex, JavaScript
- ArcGIS.com
- ArcGIS Desktop
- ArcGIS Explorer
- Mobile

- Standards
 - WMS, WCS, WMTS, KML

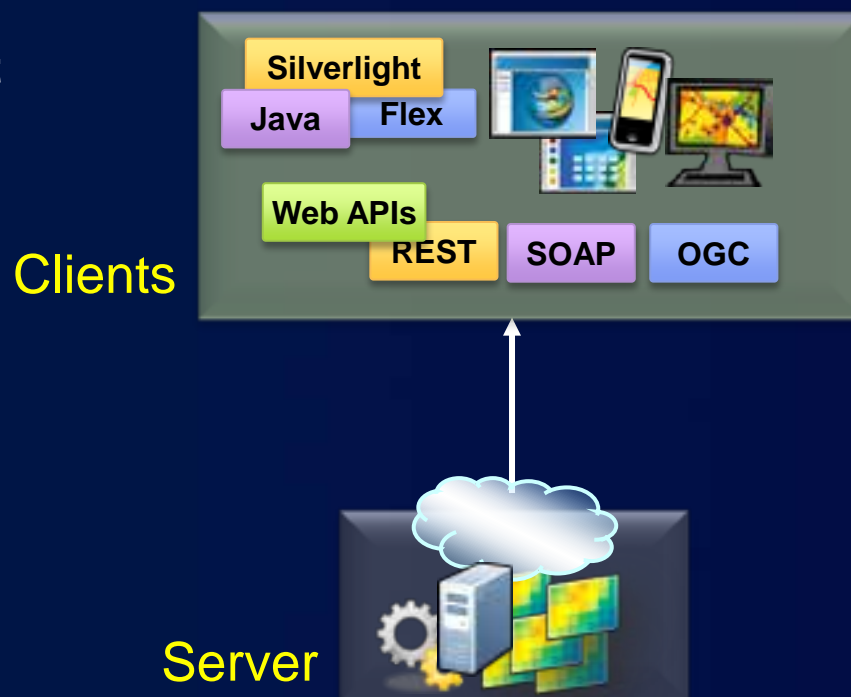


Image Service Source Data

- **Raster datasets**
 - Including raster products
- **Mosaic datasets**
 - Requires ArcGIS Server Image Extension license
- **Raster or mosaic layers**
 - To control rendering
 - Predefined query



What is the Mosaic Dataset?

- A data model used to catalog and mosaic your collections of imagery
 - Stored in geodatabase as a table
 - Each row references the image files
- On-the-fly processing and mosaicking
- Can be accessed as
 - A catalog
 - A mosaic
- Created with Geoprocessing tools



Mosaic Rules

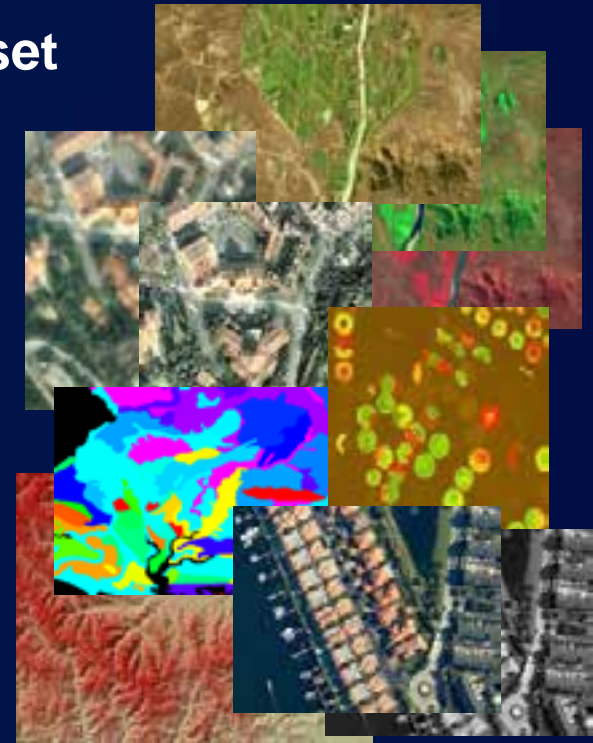
- **Mosaic Method defines the order**
 - **By Attribute**
 - **Closest to Center**
 - **Closest to Nadir**
 - **North-west (default)**
 - **Seamline**
- **Mosaic operators resolve the overlaps**
 - **First (Default)**
 - **Last/Min/Max**
 - **Mean/Blend**



Raster Function

- **Process definition stored in the dataset**
 - Image enhancement
 - Combine bands
 - Orthorectification/Pan-sharpen
 - Shaded relief, hillshade
- **Process applied as it is accessed**
- **Define on mosaic dataset or items**

- **Advantages**
 - Provide on-the-fly image processing
 - Create multiple products from one source



Publish Image Service

- **Improved desktop publishing user experience in 10.1**
 - **Support copying data during publish**
- **Suggested workflows**
 - **Server has access to the image data**
 - **Register data path and publish**
 - **Server has no access to the image data (Linux or Cloud)**
 - **move the data to the server if the data volume is large**
 - **Let publishing to copy if data volume is small**

Demo

Publish an image service

Image Service REST API

Operation	Output
ExportImage	An image in a supported format
Query ^{10.0}	A list of catalog items
Download ^{10.0}	A list of IDs
Identify ^{10.0}	Pixel values and lists of items
Measure ^{10.1}	Measurements, units, etc.
Add, Delete, Update ^{10.1}	Boolean and raster ID
ComputeHistogram ^{10.1}	An histogram

Resources: Tile, RasterCatalogItem, Colormap, RasterAttributeTable, Metadata, KeyProperties, File

Image Service Web APIs

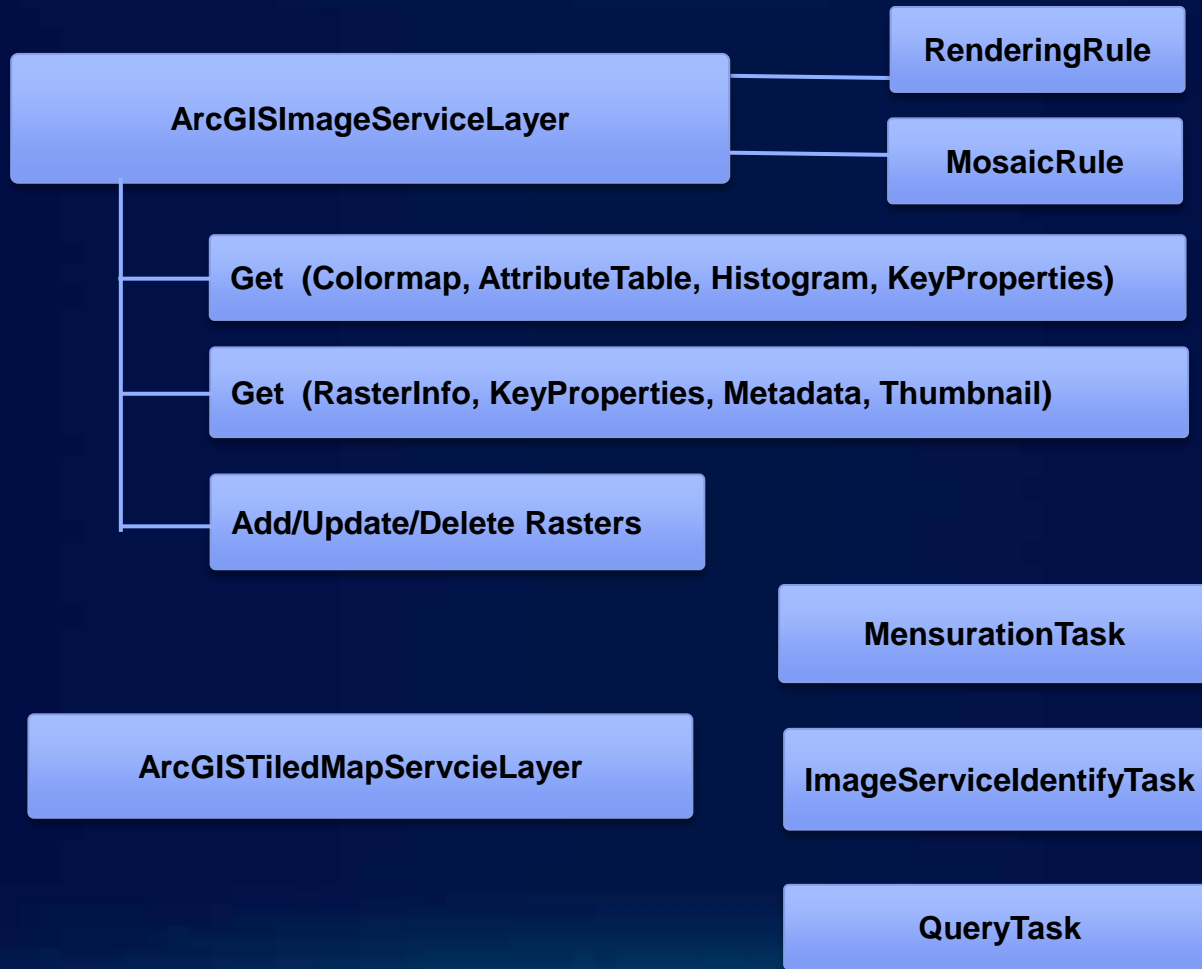


Image Service Web Application Patterns

Display
dynamic service
cached image service

Mosaic and cataloging

Dynamic processing
(server raster function)

Measure
building height
length, area, etc.

Extract pixel value
Identify
Profile

Editing service
(Add/Delete/Update)

Download/clip raw images
Export process image

Display Image Service

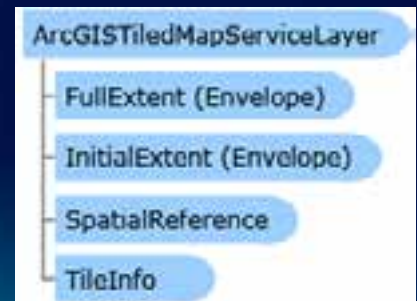
- **Dynamic image service**

```
var imageLayer = new ArcGISImageServiceLayer()
{
    ID = "Image",
    BandIds = new int[] { 2, 1, 0 },
    ImageFormat = ArcGISImageServiceLayer.
        ImageServiceImageFormat.JPGPNG,
    CompressionQuality = 75,
    Url = "http://server/arcgis/rest/services/image/ImageServer"
};
imageLayer.Initialized += new EventHandler<EventArgs>(isLayer_Initialized);
imageLayer.InitializationFailed += new EventHandler<EventArgs>(isLayer_InitializationFailed);
Map.Layers.Add(imageLayer);
```



- **Image service cache**

```
var imageLayer = new ArcGISTiledMapServiceLayer()
{
    ID = "Image",
    Url = "http://server/arcgis/rest/services/image/ImageServer"
};
```



Demo

Display dynamic and cached image services

Mosaicking and Cataloging

ArcGISImageServiceLayer

MosaicRule

- Apply to image service from mosaic dataset
- Use MosaicRule class for mosaicking
 - Set mosaic method
 - Set mosaic operator

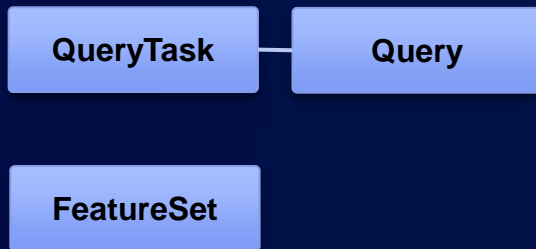
```
MosaicRule mRule = new MosaicRule()
{
    MosaicMethod = "esriMosaicAttribute",
    SortField = "YEAR",
    SortValue = 3000,
    Ascending = true,
    MosaicOperation = "MT_FIRST"
};
isLayer.MosaicRule = mRule;
```

Mosaic methods:

esriMosaicCenter
esriMosaicLockRaster
esriMosaicAttribute
esriMosaicNorthwest
esriMosaicNadir
esriMosaicSeamline
esriMosaicViewpoint

Mosaicking and cataloging (Continue)

- To access the catalog use QueryTask



```
//Define query parameter
Query query = new ESRI.ArcGIS.Client.Tasks.Query();
query.Geometry = args.Geometry;
query.Where = "Category = 1 order by Name";
query.ReturnGeometry = true;
query.OutSpatialReference = Map.SpatialReference;
query.OutFields.Add("*");

//Create a query task
QueryTask queryTask = new QueryTask(imageserviceUrl);
queryTask.ExecuteCompleted += QueryTask_ExecuteCompleted;
queryTask.Failed += QueryTask_Failed;

//Excute query task with defined parameters
queryTask.ExecuteAsync(query);
```

Demo

Mosaicking and cataloging

Server Raster Function

- **A process definition on the server that is used to response to client call**

- **On the fly processing images**
- **Support many built-in functions**

Aspect, ShadedRelief, Slope, Hillshade
Colormap, Statistic, Stretch, NDVI,
Remap, Convolution

- **Use RenderingRule class to send process request**

```
var renderingRule = new RenderingRule();  
renderingRule.RasterFunctionName = "Hillshade";  
  
var functionArguments = new Dictionary<string, object>();  
functionArguments.Add("Azimuth", 315);  
functionArguments.Add("Altitude", 45);  
functionArguments.Add("ZFactor", 1);  
  
renderingRule.RasterFunctionArguments = functionArguments;  
isLayer.RenderingRule = renderingRule;
```

ArcGISImageServiceLayer

RedenderingRule

Extending Server Raster Function

- **Expandable through raster function template**
 - **Create with ArcMap**
 - **Added to server during publishing**
 - **Support customized raster function (COM extensible)**

```
var renderingRule = new RenderingRule();
renderingRule.RasterFunctionName = "MyRFTName";

IDictionary<string, object> functionArguments =
    new Dictionary<string, object>();
functionArguments.Add("Param1", val1);
functionArguments.Add("Param2", val2);
functionArguments.Add("Param3", val3);

renderingRule.RasterFunctionArguments = functionArguments;
isLayer.RenderingRule = renderingRule;
```

Demo

Create Raster Function Template

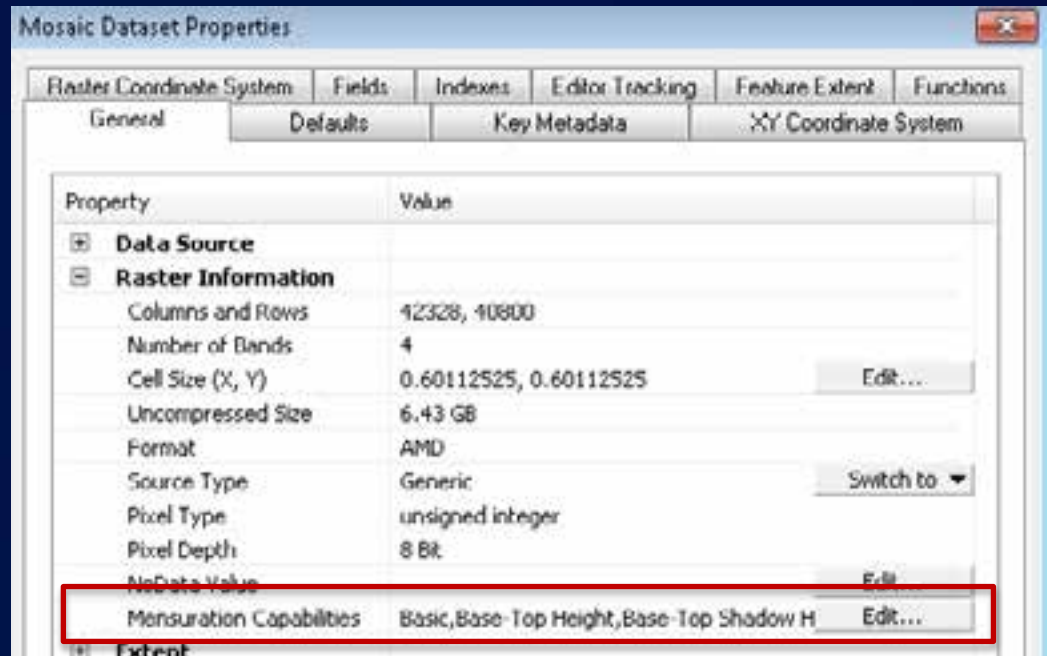
Stretch

Remap-Colormap-ColormapToRGB

Image Mensuration

- Measuring height/distance/area/location from image service

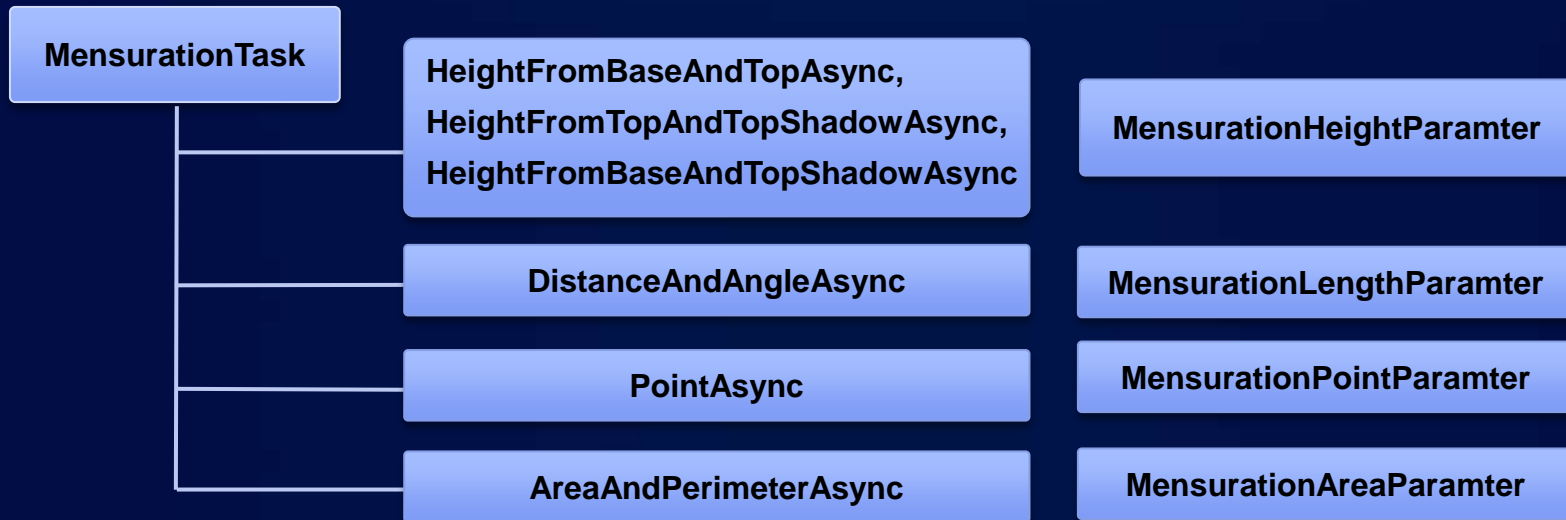
Mensuration capabilities are defined when dataset is created from image metadata



Enable mensuration capability during publishing the service



Image Mensuration API



```
var mensurationTask = new MensurationTask(isLayer.url);
var mensurationParams = new MensurationHeightParameter()
{
    MosaicRule = isLayer.MosaicRule,
    LinearUnit = esriUnits.esriMeters,
    PixelSize = new Size(1,1)
}
mensurationTask.HeightFromBaseAndTopCompleted +=
    new EventHandler<MensurationHeightEventArgs>(measureCompleted);
mensurationTask.HeightFromBaseAndTopAsync((MapPoint)fromGeometry,
    topoint, mensurationParams);
```


Demo

Measure Washington Monument

Identifying and Profiling

- **Identify pixel values on an image service using** `ImageServiceIdentifyTask`
- **Elevation profile**
 - multiple requests
- **Time-series profile**
 - identify one point, return all pixel values at this location from time-series image collection

```
var identifyTask = new ImageServiceIdentifyTask(imageserviceUrl);
var identifyParams = new ImageServiceIdentifyParameters();
identifyParams.Geometry = point;
identifyParams.MosaicRule = isLayer.MosaicRule;
identifyTask.ExecuteCompleted += identifyTask_ExecuteCompleted;
identifyTask.ExecuteAsync(identifyParams);
```

Demo

Elevation Profile
Temperature trend

Image Service Editing

- **Allow add/delete/update images from image service**
- **The backend must be an SDE mosaic dataset**
 - optional turn on editor tracking
- **Enable editing capability**
 - **Set workspace on server**
 - **Set raster type**
 - **Extend using raster type template**
 - **Example: LAS**
- **Access control**
 - AGS users can add**
 - Owner can update delete**

Operations allowed:

<input checked="" type="checkbox"/> Catalog	<input checked="" type="checkbox"/> Edit	<input checked="" type="checkbox"/> Mensuration	<input type="checkbox"/> Pixels
<input type="checkbox"/> Download	<input checked="" type="checkbox"/> Image	<input checked="" type="checkbox"/> Metadata	

Operations allowed on items added by other users

<input checked="" type="checkbox"/> Update	<input type="checkbox"/> Delete
--	---------------------------------

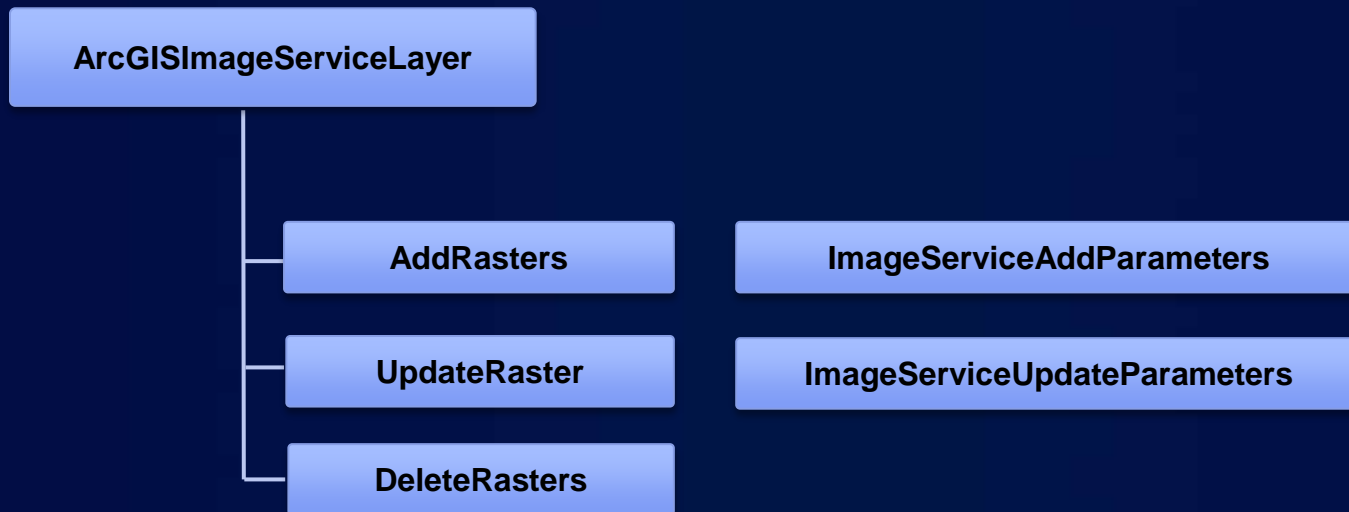
Allowed types for add

Basic types

<input checked="" type="checkbox"/> Raster Dataset	<input checked="" type="checkbox"/> WCS	<input checked="" type="checkbox"/> Image Service
<input type="checkbox"/> DTED	<input checked="" type="checkbox"/> WMS	<input type="checkbox"/> CIB
<input type="checkbox"/> CADRG/ECRG	<input checked="" type="checkbox"/> Map Service	<input type="checkbox"/> NITF

Manage custom types Manage...

Image Service Editing APIs



Demo

Uploading LAS files to an elevation service

Download and Export Image

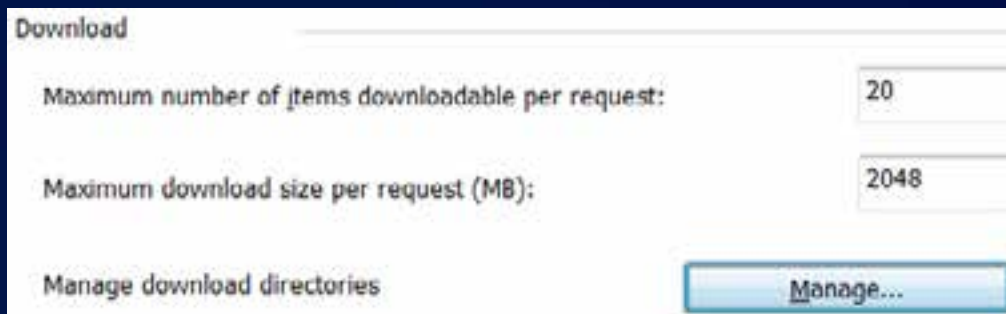
- **Download raw images from server**
 - Applies to image service of mosaic dataset
 - Clip/Convert in pixel space/original image/Maintain transformation

- **Export image**
 - Proccessed image
 - TIFF (GeoTIFF)

Downloading Capability

- **Enable catalog and download capability during publishing**

- **Download size**
- **Maximum download**
- **Download directory**



Download and ExportImage REST API

- **Download**

- RasterIds, clip extent, format
- Use File resource to retrieve rasters
- Subject to size/count limit per request

- **ExportImage**

- Define band, extent, mosaic rule, etc
- TIFF
- Subject to size limit per request (4100x15000)

Demo

Download and Export

Performance

- **Background: Cached service;**
- **Dynamic access: JPGPNG Format, or jpg**
- **Identify: use ReturnCatalogItems, ReturnGeometry flags**
- **Hybrid use: cache for display; dynamic access for query, identify, mensuration**

Help and Sample Services

- **Imagery resource center**

- <http://resourcesbeta.arcgis.com/en/communities/imagery/>
- <http://resources.arcgis.com/content/imagery/10.0/about>

- **Web API**

- <http://resourcesbeta.arcgis.com/en/webapis/community>
- <http://resources.arcgis.com/content/web/web-apis>

- **World Landsat service**

<http://www.esri.com/landsat>

- **World elevation services beta**

http://resources.arcgis.com/content/imagery/10.0/world_elevation

Q&A

Thank you!

Please fill out the survey form