

# 2011 Esri Developer Summit

Palm Springs, CA

## Working with image services

Melanie Harlow

Wenxue Ju

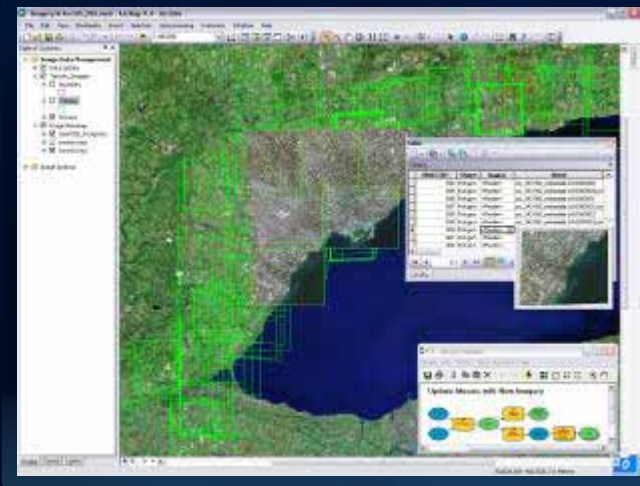


## What is an image service?

- It is imagery or raster data made available by a server to a client application

# What can you do with an image service?

- Use it as an image (visual analysis)
- Use it as raster data (pixel analysis)
- Access it as a catalog <sup>\*MD</sup>



# How can you access an image service?

- **ArcGIS Desktop**
- **ArcGIS Explorer**
- **Web APIs (Silverlight, Flex, JavaScript)**
- **ArcGIS.com**
- **REST, SOAP**
- **WMS, WCS, KML**
- **3rd Party Applications**

# Image service versus map service

Image service	Map service
Serves imagery (raster data) directly	Serves a map document containing imagery or vector data
Layer properties can be altered by client <ul style="list-style-type: none"><li>• Compression</li><li>• Rendering</li><li>• Band combinations</li></ul>	Client views map service as it was designed  Client cannot change layer properties
Can be saved as a layer and used as a data source	Can build cache

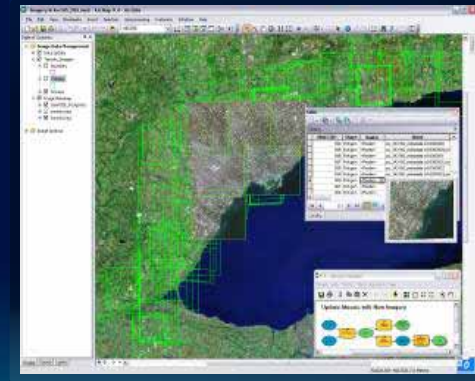
# Image service source data

- **Data sources**
  - Raster datasets
  - **Mosaic datasets** *new at 10.0*
    - Requires ArcGIS Server Image Extension
  - **Raster or mosaic layers**
    - To control rendering
    - Preset some layer properties
    - Predefined query



# What is the mosaic dataset?

- **A new geodatabase data model used to catalog and process your collections of imagery**
  - Stored as a table and viewed as a table or image
- **Indirect pixel management**
  - Images can remain in their native format on disk or be loaded into the geodatabase
- **Unlimited size\***
- **Provides dynamic mosaicking and on-the-fly processing**



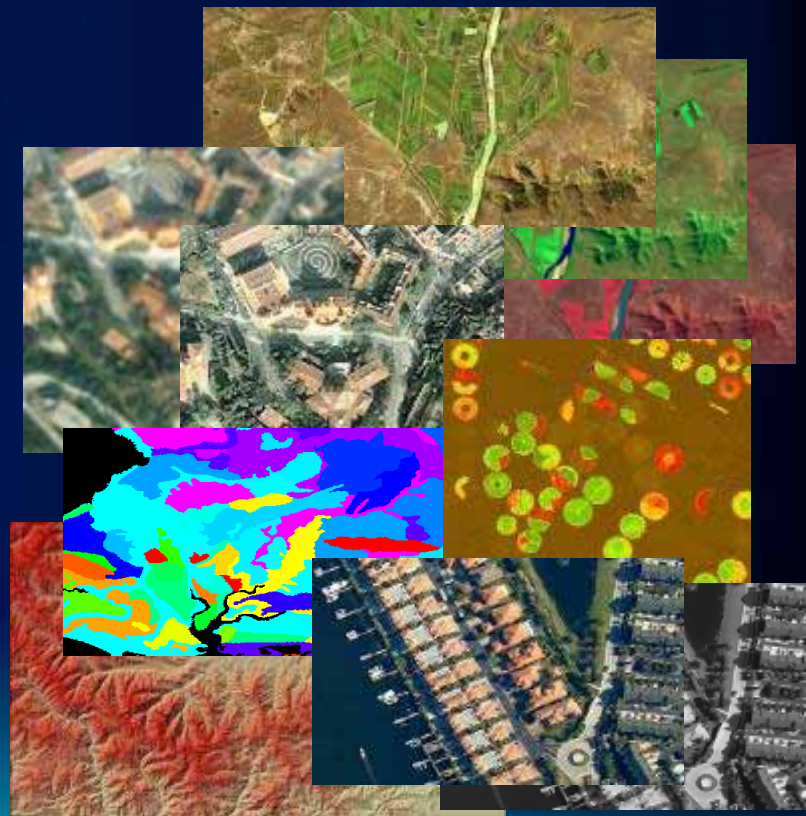
# Mosaic dataset: dynamic mosaicking

- **Overlapping imagery has higher value**
- **Footprints**
  - Used to clip imagery and remove unwanted areas
  - Vector features (Editable)
- **Queries**
  - Refine selection of imagery
- **Mosaic rules**
  - Order of displayed imagery
  - Ensure best imagery is on top
  - Can be controlled by user



# Mosaic dataset: on-the-fly processing

- Imagery processed as it is accessed
- Can create multiple products from one source
- Processing
  - Image enhancement
  - Combine bands
  - Color correction
  - Orthorectification
  - Pan-sharpen
  - Shaded relief, hillshade
- Define processing functions
  - On each raster dataset
  - On a mosaic dataset

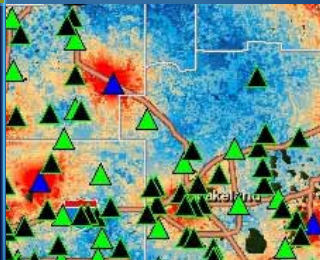
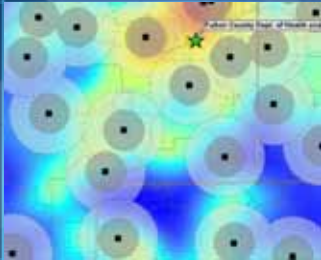


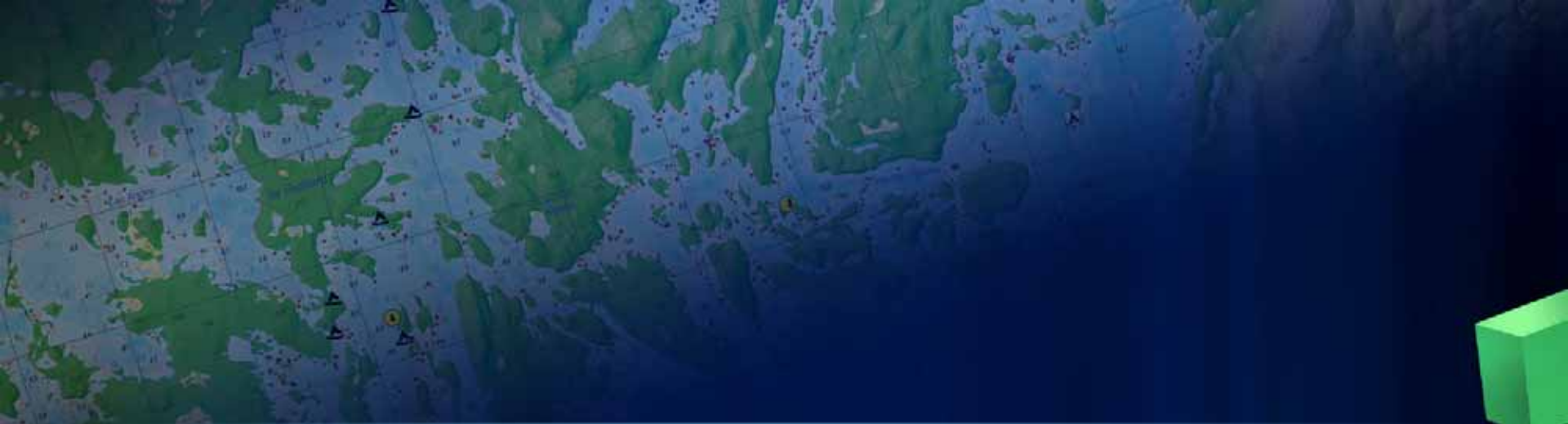
# Building a mosaic dataset

- **Simple workflow**
  1. **Create mosaic dataset**
  2. **Add imagery**
  3. **Optionally, edit properties and functions**
- **Store in a geodatabase**
  - **Build with geoprocessing tools**
  - **Automation with models or Python**
- **Can interactively edit and view in ArcMap**
  - **All layers are displayed**
  - **Edit and add fields in table window**



# Demo: Web application using image service





# Publishing



# Publishing an image service

- **Same process as publishing any service with ArcGIS Server**
  - **via ArcCatalog**
  - **via Server Manager**
- **Properties unique to image services**
  - **Capabilities**
  - **Parameters**
- **Some capabilities are specific to the data**

# ArcGIS Server 10 Image Extension

- **A license (not a separate install)**
- **It extends the capabilities to serve mosaic datasets**
- **Image services accessible as:**
  - **Image**
    - **Dynamic mosaicking; On-the-fly processing; Identify with metadata; Export**
  - **Catalog**
    - **Geometry and attributes; Query; Selection; Preview; Download, with clipping**

Serving:

- Rasters
- Raster layers
- \*Mosaic datasets
- \*Mosaic dataset layers



ArcGIS Server



# Image service capabilities



- Limits what users can do with your Web service

---

<b>Image</b>	<b>Allows the image to be displayed (Absorbed the Mosaic operation in 9.3x)</b>
Metadata	Allows the client to see metadata for the image service and for each raster in a mosaic dataset
Catalog	Allows the client to open the table of a mosaic dataset when it is served
Download	Allows rasters to be downloaded when serving a mosaic dataset
Pixels	Allows the API developer to access the pixel blocks of the individual rasters in a mosaic dataset

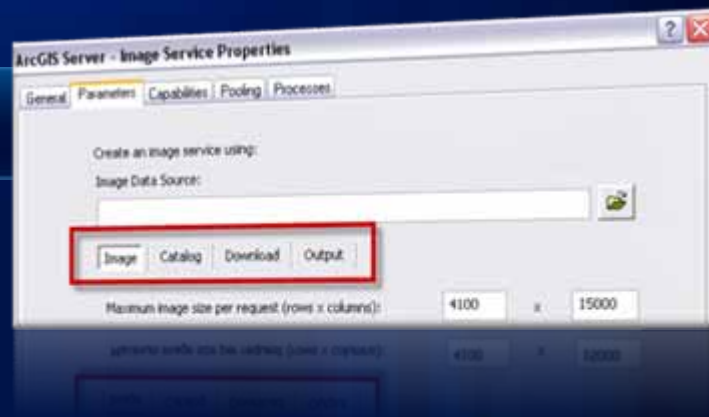
---

# OGC Capabilities

- **Web Coverage Service (WCS)**
  - Serves actual pixel information from source data
  - Supported in many image analysis and processing software packages
  - Useful for image analysis applications where full pixel depth (bits and bands) are required
  - NEW: Query, Time, Limit exporting
  
- **Web Map Service (WMS)**
  - Supported for “rendered” image services
  - Useful for imagery base maps



# Image Service Parameters



- Control the image data
- Image<sup>new</sup>, Catalog<sup>new</sup>, Download<sup>new</sup>, and Output
- These parameters are not applicable to all image services and depend on the input for the image service
- Many are set when you create the mosaic dataset
- Some can be altered by the server administrator
- Server administrator cannot exceed limitations
  - Such as: allowed mosaic methods, the maximum number of rasters per mosaic

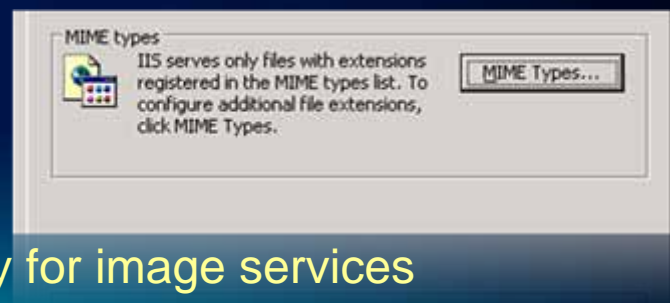
# Image Service Parameters: Image

- Affects the image that users will see

	RD	MD	Layers
Max image size per request	✓	✓	✓
Default resampling method	✓	✓	✓
Compression methods and quality	✓	✓	✓
Max # of rasters per mosaic		✓	
Allowed mosaic methods		✓	

# Image Service Parameters: Download

- **Affect the load on the server and the access users have to the source images**
- **Mosaic dataset only**
  - **Maximum number of items downloadable per request**
  - **Directories**
    - **The list of physical directories where the images to be downloaded are located**
    - **Important for the REST download**
  - **Virtual directory: where the image URLs will be mapped**
- **Define your MIME**
  - **Part of your IIS properties**
  - **Add your extensions to the list**



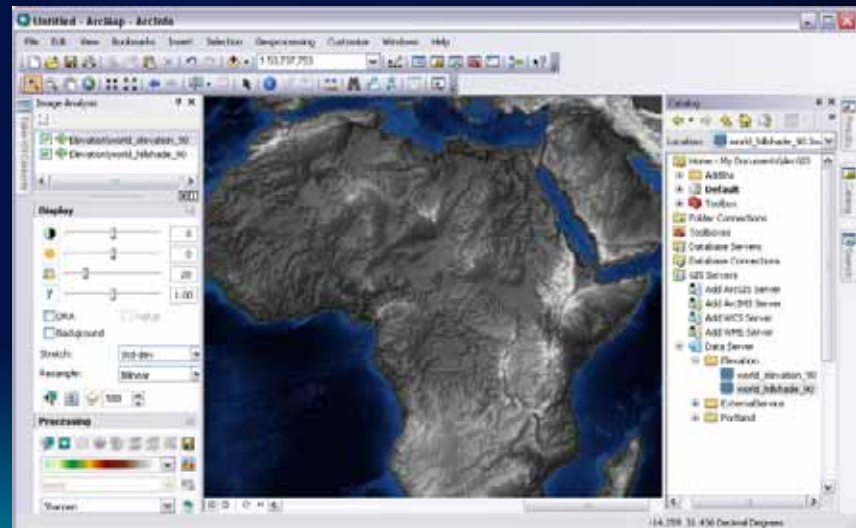
\* **Blog: Configuring the download capability for image services**

# Using image services in desktop and mobile applications

```
function getDriverId(papolystrcode)  
{  
  var map = new esri.Map(  
    "F:\WebApps\services\car  
    \shtml", {  
      services: ESRI, timeout:  
      ESRI, map.addLayer(  
        new dojo.Color(0, 0, 0, 0.5)  
      )  
    }  
  );  
  var features = papolystrcode  
  for (var f=0; f<features.length; f++)  
  {  
    f.fid = 0  
    var polysymbol =  
    new dojo.Symbol({  
      color: "red", fill:  
      "red", stroke: "red",  
      type: "poly", weight:  
      1, symbol:  
      "circle", size:  
      1000000000000000  
    }, f)  
    map.addLayer(new esri.LayerDefinition({  
      url: "http://localhost:6080/arcgis/rest/services/ESRI/ImageService",  
      type: "Image",  
      opacity: 0.5, noFrame: true, transparent: true, style: "simple",  
      symbol: polysymbol, styleClass: "ESRIImageService"}, f))  
  }
```

# Using image services in desktop applications

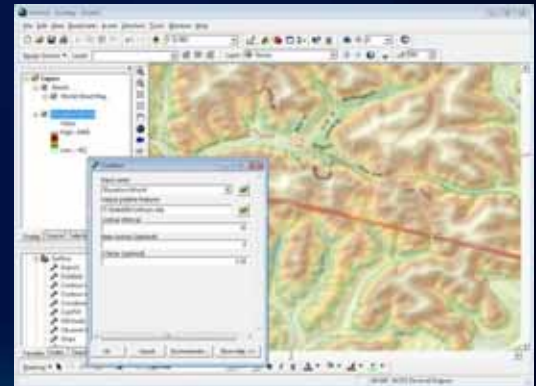
- **ArcGIS**
  - ArcMap, ArcGlobe, ArcCatalog
  - Explorer
- **Google Earth (using KMZ/KML)**
- **Gaia (OGC services)**



# Using image services in geoprocessing

- Image services can be used in geoprocessing tools as raster data
- Use Make Image Server Layer tool when:
  - Using an image service in a model or script
  - To define an extent, cell size, mosaic method, etc.
  - When using it within a geoprocessing service

[Blog: Parts 1 & 2:](#)  
Designing a geoprocessing service  
to work with an image service



## Using image services in ArcGIS Explorer

- **ArcGIS Explorer 1500 and Explorer Online has direct support for image services**
- **ArcGIS Explorer 900/1200 has support for image services via \*.lyr files**
- **Via globe services**

# Using image services in mobile applications

- **Mobile applications make live connections to image services**
- **If not live**
  - **Export image from image service**
  - **Use map cache containing the image service (in mxd)**







**Demo: Publishing and accessing  
image services in Desktop**



# Using image services in Web applications

```
function onMapClick(event) {  
    var map = new esri.Map("map", {  
        mapService: "http://services.arcgis.com/  
        XXXX/arcgis/rest/services/  
        XXXX/ESRI_TopoMap/  
        MapServer",  
        map.addLayer(layer);  
    });  
    function getDriverName(polygon) {  
        var features = result.features;  
        for (var f=0, feature; f<features.length; f++) {  
            if (feature.geometry.equals(polygon)) {  
                return feature.attributes.driverName;  
            }  
        }  
        return null;  
    }  
    do {  
        Symbol = new esri.Symbol({color: "#FF0000", style: "circle", size: 100});  
        polygon = new esri.Polygon([  
            [100, 100], [100, 200], [200, 200], [200, 100], [100, 100]  
        ]);  
        feature = getDriverName(polygon);  
        if (feature) {  
            map.addLayer(new esri.FeatureLayer(feature, Symbol));  
        }  
    } while (true);  
}
```

# Image services and Web applications


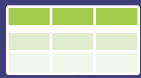

- Supported through SOAP, REST, WMS, WCS, KML
- Accessible in ArcGIS Online – My Map
- Accessible through ArcGIS Server APIs
  - JavaScript
  - Flex
  - Silverlight
- Flex viewer
- Silverlight viewer (coming)



# Using image services in Web applications

- **Same display qualities of Desktop applications**
- **REST and SOAP support modifying all image service layer properties**
  - **Transmission compression**
  - **Band combinations**
  - **Mosaic methods and properties**
  - **Resampling**
- **Use in geoprocessing services**

# Image Service REST API

Sources	Operations	Results
Mosaic dataset Raster dataset	ExportImage	 An image in a supported format
Mosaic dataset	Query <sup>10.0</sup>	 A list of catalog items
Mosaic dataset	Download <sup>10.0</sup>	 A list of file IDs for download
Mosaic dataset Raster dataset	Identify <sup>10.0</sup>	Pixel values and lists of catalog items

\* Next release: Upload, Mensuration

# ExportImage Enhancements

- Specifies how and what to get from the server
- bbox, size, SR, pixel type, compression, format, etc.

- **MosaicRule**

```
{ "mosaicMethod" : "esriMosaicLockRaster", "lockRasterIds" : [32, 454, 14] }
```

- **RenderRule**

```
{ "rasterFunction" : "Slope",  
  "rasterFunctionArguments" :  
    { "ZFactor" : 0.3 },  
  "variableName" : "DEM" }
```

## Supported raster functions:

Aspect	ShadedRelief
Colormap	Hillshade
NDVI	Statistic
Slope	Stretch

\* Next release: More functions

# ExportImage Enhancements: Time

- Time

**Syntax** =<timeInstant>

Example: time=1199145600000

(1 Jan 2008 00:00:00 GMT)

**Syntax:** time=<startTime>, <endTime>

Example: time=1199145600000, 1230768000000

(1 Jan 2008 00:00:00 GMT to 1 Jan 2009 00:00:00 GMT)

## ExportImage Enhancements: Image Format

- jpgpng | png | png8 | png24 | jpg | bmp | gif | tiff
- **JPGPNG** returns a **JPG** if there are no transparent pixels in the requested extent, otherwise it returns a **PNG**.
  - **JPG** – smaller & faster
  - **PNG** – provides transparency
  - **TIFF** – geotiff (SP1)
- **Syntax example: format=png24**



## REST address example

<http://<myserver>/ArcGIS/rest/services/<MyImage>/ImageServer/exportImage?>

<http://sampleserver3.arcgisonline.com>

## REST address example

<http://<myserver>/ArcGIS/rest/services/<MyImage>/ImageServer/exportImage?>

<http://sampleserver3.arcgisonline.com/ArcGIS/rest/services>

## REST address example

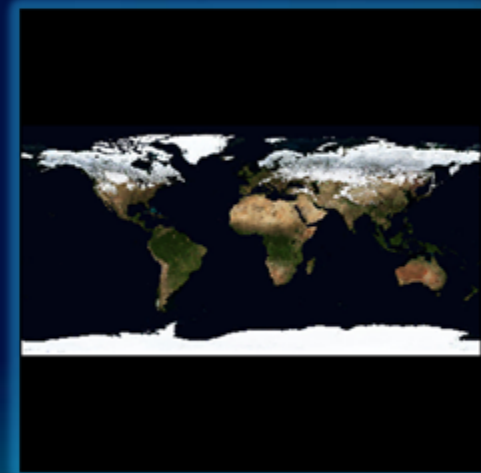
<http://<myserver>/ArcGIS/rest/services/<MyImage>/ImageServer/exportImage?>

<http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/World/MODIS/ImageServer>

## REST address example

<http://<myserver>/ArcGIS/rest/services/<MyImage>/ImageServer/exportImage?>

<http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/World/MODIS/ImageServer/exportImage?>



## REST address example

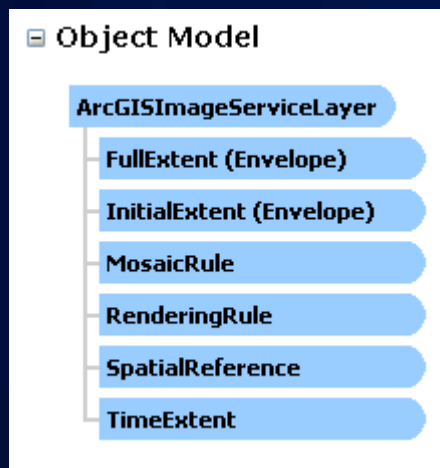
<http://<myserver>/ArcGIS/rest/services/<MyImage>/ImageServer>

[http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/World/MODIS/ImageServer/exportImage?bbox=115,34,146,31&bboxSR=4326&format=jpg&compressionQuality=85&pixelType=U8&interpolation=RSP\\_NearestNeighbor](http://sampleserver3.arcgisonline.com/ArcGIS/rest/services/World/MODIS/ImageServer/exportImage?bbox=115,34,146,31&bboxSR=4326&format=jpg&compressionQuality=85&pixelType=U8&interpolation=RSP_NearestNeighbor)



# ArcGISImageServiceLayer Class

- Allows you to work with an image service resource exposed by the ArcGIS Server REST API



## Demo: ArcGISImageServiceLayer

```
ArcGISImageServiceLayer isLayer = new ArcGISImageServiceLayer();  
isLayer.Url = imageserviceUrl;  
Map.Layers.Add(isLayer);
```

# Query

- **Uses generic QueryTask class**
- **Attribute query**
- **Spatial query**
- **Time query**



## Demo: Query

```
QueryTask queryTask = new QueryTask(queryurl);  
Query query = new Query();  
query.Geometry = args.Geometry;  
query.Where = "Category = 1";  
queryTask.ExecuteAsync(query);
```

# Identify

- **Uses ImageServiceIdentifyTask class**
- **Takes:**
  - **Geometry (point or polygon)**
  - **Mosaic method rule**
  - **Pixel size**
- **Returns**
  - **Value of the pixel (can be per image)**
  - **Associated attributes per image (catalog table)**

## Demo:Identify

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

# Download

- **Get raw data**
  - Transmit source dataset and ancillary files
- **Export the processed result**
  - Save/As (data export) to jpg/tiff, etc
  - But in Web use Export Image
    - Mostly ungeoreferenced, In 10.0 SP1 the tiff is a geotiff
- **Make REST call (WebRequest)**

**Demo: Download**

# **Demo: API References Online and Image Gallery Samples**

# Summary

- **What is an image service**
- **Publishing**
- **Using**
  - **Desktop applications**
  - **REST**
  - **Web applications**



esri