



ArcPy: Beyond the basics of arcpy.mp

Jeff Barrette and Jeff Moulds



Jeff Barrette
Esri – Olympia, WA



Jeff Moulds
Esri – Redlands, CA

2021 ESRI
DEVELOPER SUMMIT

Python CIM Access in Pro

Introduced with ArcGIS Pro 2.4

- CIM (Cartographic Information Model)
 - Specification for how project and document information is persisted and re-created
 - MAPX, PAGX, LYRX files are in a JSON format
 - Structure is analogous to an object model diagram
- Arcpy.mp can navigate the CIM object model
 - Entry points are with the Map, Layer, Table or Layout objects

```
"featureTable" : {  
  "type" : "CIMFeatureTable",  
  "displayField" : "NAME",  
  "editable" : true,  
  "fieldDescriptions" : [  
    {  
      "type" : "CIMFieldDescription",  
      "alias" : "OBJECTID",  
      "fieldName" : "OBJECTID",  
      "numberFormat" : {  
        "type" : "CIMNumericFormat",  
        "alignmentOption" : "esriAlign",  
        "alignmentWidth" : 0,  
        "roundingOption" : "esriRoundN",  
        "roundingValue" : 0  
      },  
      "readOnly" : true,  
      "visible" : true,  
      "searchMode" : "Exact"  
    },  
  ],  
}
```

```
cim_lyr = lyr.getDefinition('V2')  
cim_lyr.featureTable.fieldDescriptions[0].visible = False
```

Why CIM Access ... and why not

- Provide finer grained access to project properties
- Pro is growing so rapidly, the CIM immediately exposes new capabilities
- It has been available to the .NET SDK community since Pro 1.1
- Why not use the CIM
 - Doesn't provide access to everything
 - high level project properties, metadata, change spatial reference
 - Can't create new objects. If it is not in the CIM, you can't access it.
 - No life guard (managed UI or API) on duty
 - You can break the behavior of the app
 - Test thoroughly



Basic workflow

```
cim_obj = Object.getDefinition(version) #'V2'
```

```
### Do something to the CIM object
```

```
Object.setDefinition(cim_obj)
```

* **Object** = Map, Layer, Table or Layout


```
"featureTable" : {  
  "type" : "CIMFeatureTable",  
  "displayField" : "NAME",  
  "editable" : true,  
  "fieldDescriptions" : [  
    {  
      "type" : "CIMFieldDescription",  
      "alias" : "OBJECTID",  
      "fieldName" : "OBJECTID",  
      "numberFormat" : {  
        "type" : "CIMNumericFormat",  
        "alignmentOption" : "esriAlignRight",  
        "alignmentWidth" : 0,  
        "roundingOption" : "esriRoundNumberOfDecimals",  
        "roundingValue" : 0  
      },  
      "readOnly" : true,  
      "visible" : true,  
      "searchMode" : "Exact"  
    },  
  ],  
}
```

```
cim_lyr = lyr.getDefinition('V2')  
cim_lyr.featureTable.fieldDescriptions[0].visible = False
```

Python CIM Access

Demo: JSON object model demo

Jeff Barrette



CIM Samples:

<https://esriurl.com/8899>

Creating “CIM” objects

Introduced with ArcGIS Pro 2.5

```
RGBColor = arcpy.cim.CreateCIMObjectFromClassName('CIMRGBColor', 'V2')
RGBColor.values = [130, 130, 130, 100]
m_cim.backgroundColor = RGBColor
```

```
27 "layers" : [
28     "CIMPATH=map/greatlakes.xml",
29     "CIMPATH=map/new_group_layer.xml",
30     "CIMPATH=e49b3ea0ed704bf88806074a3b3aea37.xml",
31     "CIMPATH=f7bea83c2d7d40c298ec29c6a6207ede.xml"
32 ],
33 "defaultViewingMode" : "Map",
34 "mapType" : "Map",
35 "bookmarks" : [
36     {
37         "type" : "CIMBookmark",
38         "name" : "Lakes",
39         "thumbnailImagePath" :
40         "CIMPATH=Thumbnails/ade916a8f93f9cd635b0afce218758e
41         .dat",
42         "camera" : {
43             "type" : "CIMViewCamera",
44             "pitch" : -90,
```

```
27 "layers" : [
28     "CIMPATH=map/greatlakes.xml",
29     "CIMPATH=map/new_group_layer.xml",
30     "CIMPATH=e49b3ea0ed704bf88806074a3b3aea37.xml",
31     "CIMPATH=f7bea83c2d7d40c298ec29c6a6207ede.xml"
32 ],
33 "defaultViewingMode" : "Map",
34 "mapType" : "Map",
35 "backgroundColor" : {
36     "type" : "CIMRGBColor",
37     "values" : [
38         130,
39         130,
40         130,
41         100
42     ]
43 },
44 "bookmarks" : [
```

```
# - If a color does NOT exist, you must use the
# CreateCIMObjectFromClassName function to create the appropriate
# CIM color object. For simplicity, you can create the color
# object and apply it without needing an if statement but the
# script shows how to test for None to avoid a runtime error.
# - You might have to turn off layers to see the underlying
# background color change.

p = arcpy.mp.ArcGISProject('current')
m = p.listMaps('GreatLakes')[0]

m_cim = m.getDefinition('V2')          #Get the Map's CIM definition

#Check to see if a color exists, if not create a color
if m_cim.backgroundColor is None:
    RGBColor = arcpy.cim.CreateCIMObjectFromClassName('CIMRGBColor', 'V2')
    RGBColor.values = [130, 130, 130, 100]
    m_cim.backgroundColor = RGBColor

#If a color does exist, simply change its values
else:
    m_cim.backgroundColor.values = [80, 80, 80, 100]

m.setDefinition(m_cim)                #set the Map's CIM definition
```

Python CIM Access

Demo: Creating CIM Objects

Jeff Barrette



Creating new objects can take some work

You have to “reverse” engineer the object when objects have dependencies on other objects

```
LineSymLyr1 = arcpy.cim.CreateCIMObjectFromClassName('CIMSolidStroke', 'V2')
LineSymLyr1.color = LineRGBColor
LineSymLyr1.capStyle = "Square"
LineSymLyr1.joinStyle = "Miter"
LineSymLyr1.width = 4

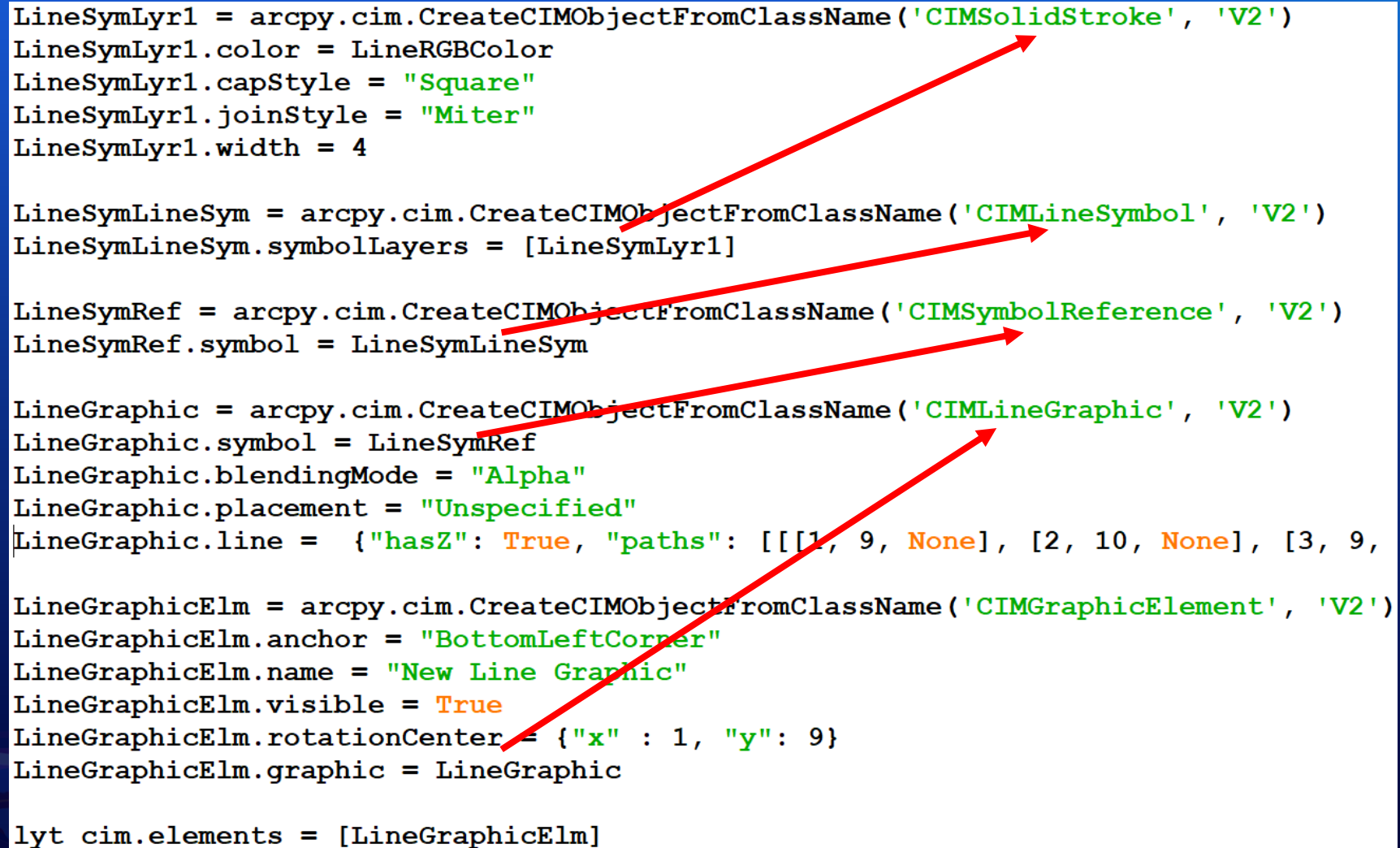
LineSymLineSym = arcpy.cim.CreateCIMObjectFromClassName('CIMLineSymbol', 'V2')
LineSymLineSym.symbolLayers = [LineSymLyr1]

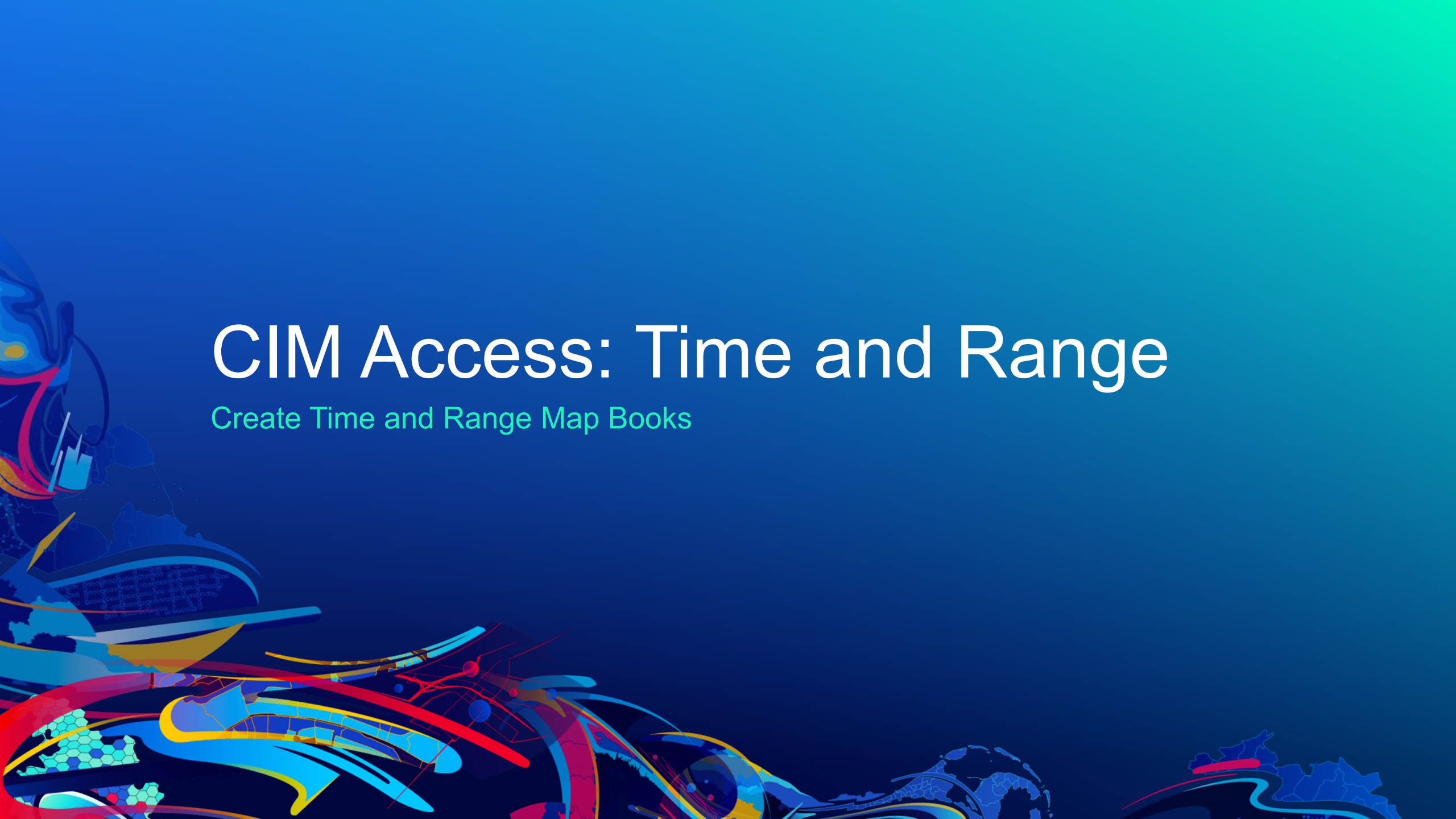
LineSymRef = arcpy.cim.CreateCIMObjectFromClassName('CIMSymbolReference', 'V2')
LineSymRef.symbol = LineSymLineSym

LineGraphic = arcpy.cim.CreateCIMObjectFromClassName('CIMLineGraphic', 'V2')
LineGraphic.symbol = LineSymRef
LineGraphic.blendingMode = "Alpha"
LineGraphic.placement = "Unspecified"
LineGraphic.line = {"hasZ": True, "paths": [[[1, 9, None], [2, 10, None], [3, 9,

LineGraphicElm = arcpy.cim.CreateCIMObjectFromClassName('CIMGraphicElement', 'V2')
LineGraphicElm.anchor = "BottomLeftCorner"
LineGraphicElm.name = "New Line Graphic"
LineGraphicElm.visible = True
LineGraphicElm.rotationCenter = {"x" : 1, "y": 9}
LineGraphicElm.graphic = LineGraphic

lyt_cim.elements = [LineGraphicElm]
```

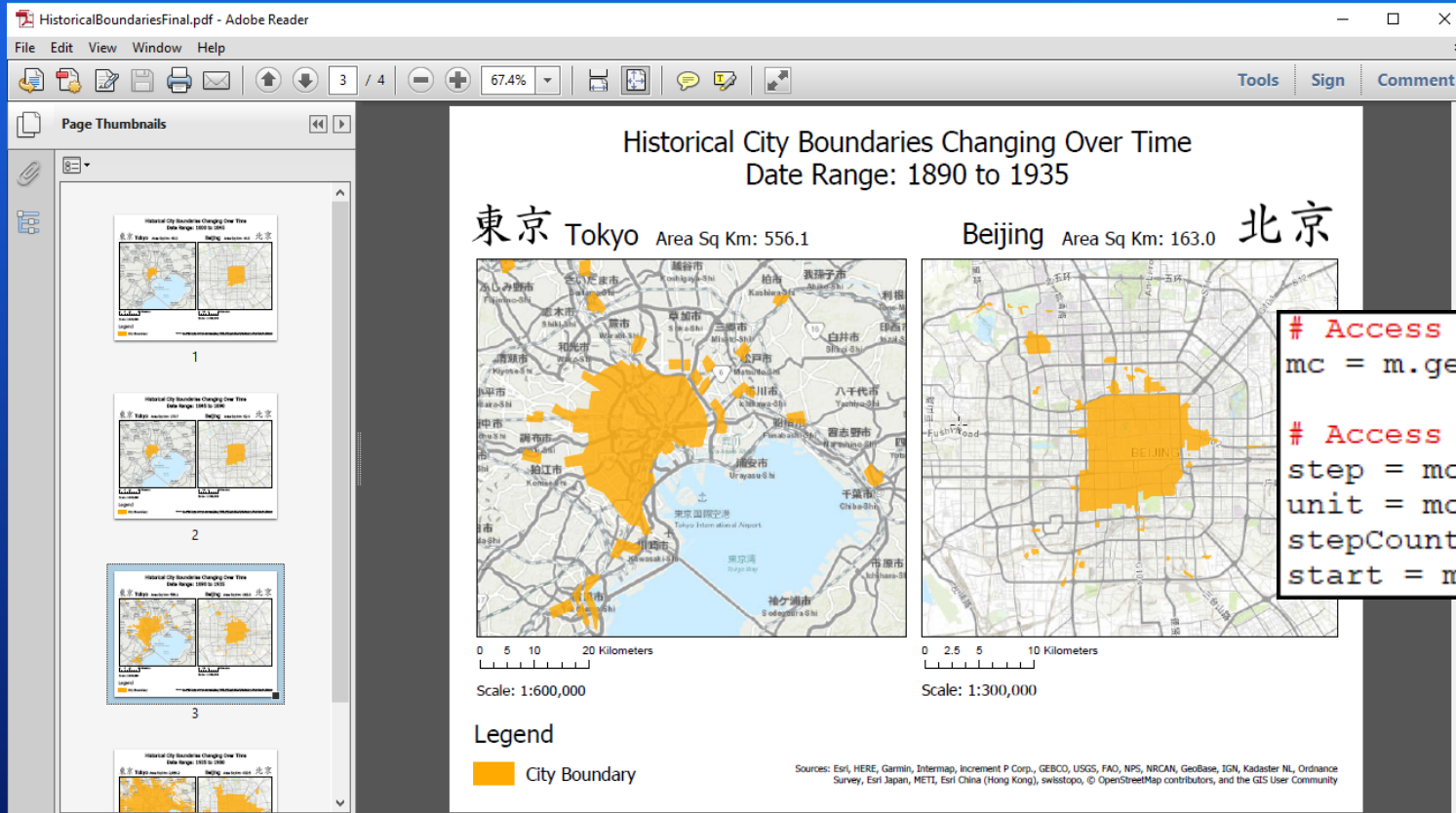




CIM Access: Time and Range

Create Time and Range Map Books

Time Map Book



```
# Access map cim
mc = m.getDefinition("v2")

# Access cim time properties
step = mc.timeSliderSettings.stepIntervalValue
unit = mc.timeSliderSettings.stepIntervalUnit
stepCount = mc.timeSliderSettings.stepCount
start = mc.timeDisplay.currentTimeExtent.start
```

Time Map Book demo: <https://www.youtube.com/watch?v=RkyzeOgCHe4>

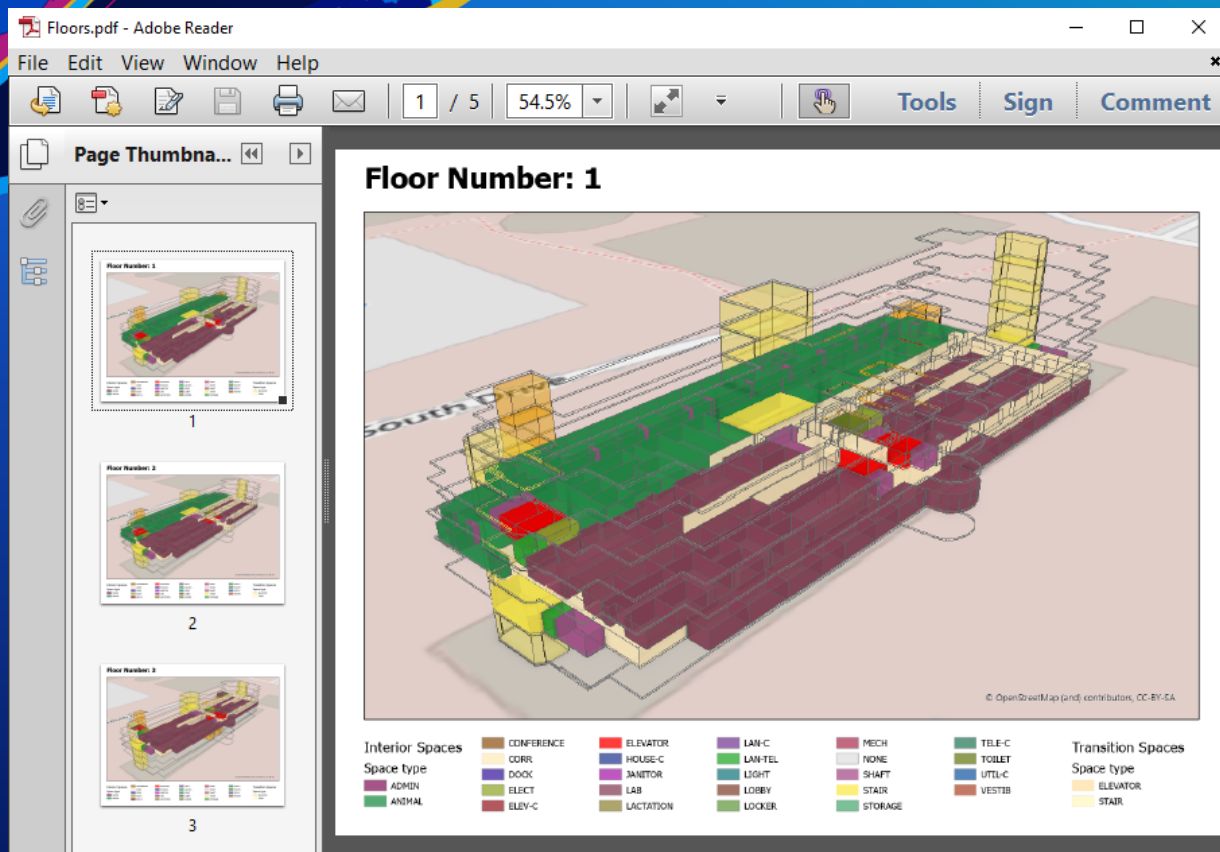
ArcGIS Pro 2.8: Proposed MapTime and LayerTime classes

- **LayerTime Class**

- enableTime(...)
- daylightSavings
- endTime
- endTimeField
- startTime
- startTimeField
- timeFormat
- timeOffset
- timeOffsetUnits
- timeStepInterval
- timeStepIntervalUnits
- timeZone

- **MapTime Class**

- timestep(...)
- currentTimeStart
- currentTimeEnd
- currentTimeSpan
- currentTimeSpanUnits
- isTimeEnabled
- timeStepInterval
- timeStepIntervalUnits
- timeInclusion



Range Map Book Demo

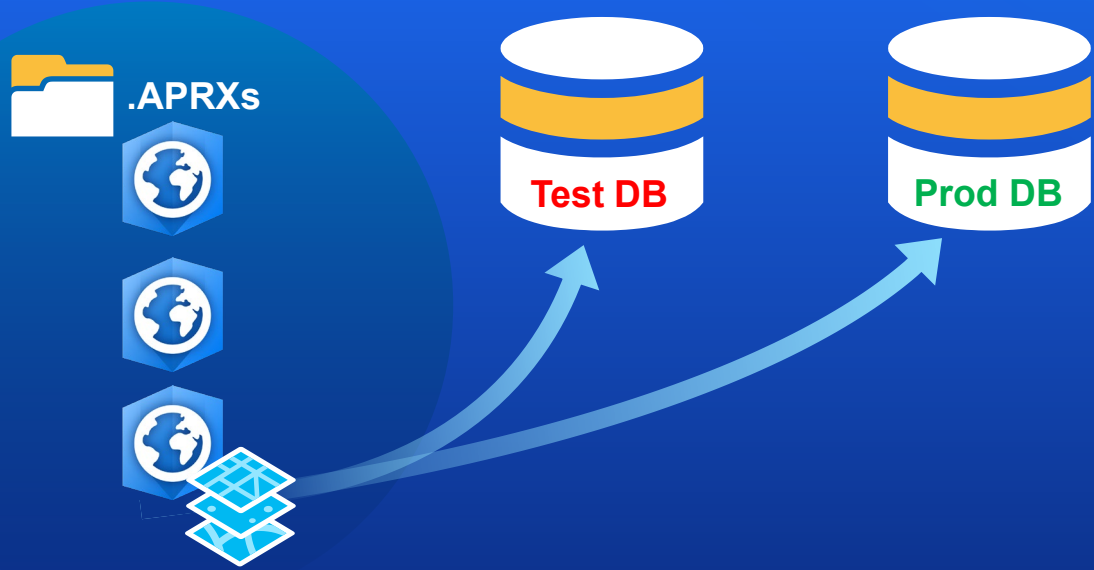
Presenter: Jeff Moulds



CIM Access: Data Sources

Using arcpy CIM to update data sources

Updating Data Sources in ArcGIS Pro with arcpy.mp



- Updating data sources arcpy help topic <http://esriurl.com/13936>
- Updating data sources UI help topic <http://esriurl.com/10000>

Find this path:

```
aprx = arcpy.mp.ArcGISProject(r'C:\Projects\YosemiteNP\Yosemite.aprx')  
aprx.updateConnectionProperties(r'C:\Projects\YosemiteNP\DBConnections\TestGDB.sde',  
                               r'C:\Projects\YosemiteNP\DBConnections\ProductionGDB.sde')
```

Replace it with this path:

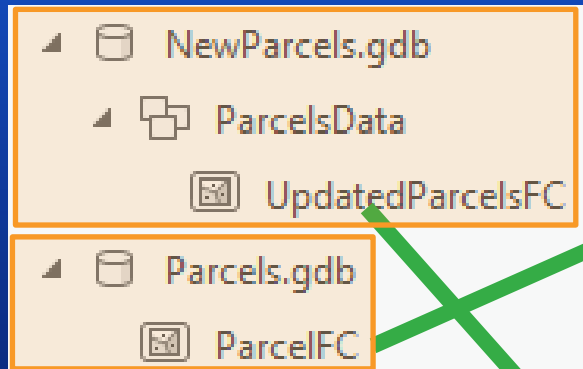
CIM: Updating data sources and modifying feature datasets

Existing data:

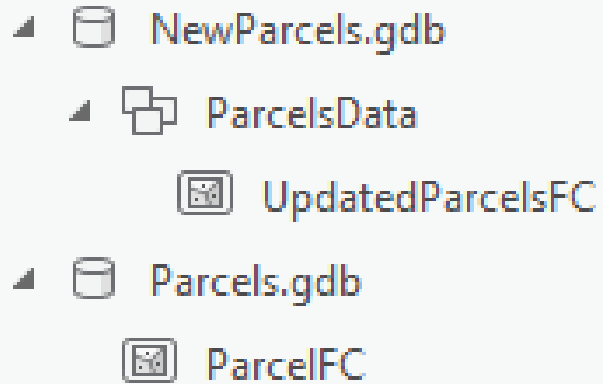
```
54 "featureTable" : {  
55   "type" : "CIMFeatureTable",  
56   "displayField" : "Shape_Length",  
57   "editable" : true,  
58   "dataConnection" : {  
59     "type" : "CIMStandardDataConnection",  
60     "workspaceConnectionString" : "DATABASE=C:\\Projects\\Data\\Parcels.gdb",  
61     "workspaceFactory" : "FileGDB",  
62     "dataset" : "ParcelFC",  
63     "datasetType" : "esriDTFeatureClass"  
64   },  
65   "studyAreaSpatialRel" : "SpatialRelUndefined"
```

New data:

```
53   "elevationExpression" : "Shape_Length"  
54 "featureTable" : {  
55   "type" : "CIMFeatureTable",  
56   "displayField" : "Shape_Length",  
57   "editable" : true,  
58   "dataConnection" : {  
59     1 "type" : "CIMFeatureDatasetDataConnection",  
60     2 "featureDataset" : "ParcelsData",  
61     3 "workspaceConnectionString" : "DATABASE=C:\\Projects\\Data\\NewParcels.gdb",  
62     "workspaceFactory" : "FileGDB",  
63     "dataset" : "UpdatedParcelsFC",  
64     "datasetType" : "esriDTFeatureClass"  
65   },  
66   "studyAreaSpatialRel" : "SpatialRelUndefined"
```



CIM: Updating data sources and modifying feature data sets



```
1 # Specify the new geodatabase properties
2 newGDB = r'C:\Projects\Data\NewParcels.gdb'
3 newFeatureClass = "UpdatedParcelsFC"
4
5 # Reference project, map and layer
6 p = arcpy.mp.ArcGISProject(r'C:\Projects\YosemiteNP\Yosemite.aprx')
7 m = p.listMaps('Parcels Map')[0]
8 l = m.listLayers('Parcels')[0]
9
10 # Get the layer's CIM definition
11 lyrCIM = l.getDefinition('V2')
12
13 # Create a new CIM data connection
14 dc = arcpy.cim.CreateCIMObjectFromClassName('CIMStandardDataConnection', 'V2')
15
16 # Specify the geodatabase
17 dc.workspaceConnectionString = f"DATABASE={newGDB}"
18
19 # Specify the workspace type
20 dataSource.workspaceFactory = "FileGDB"
21
22 # Specify the dataset name
23 dc.dataset = newFeatureClass
24
25 # Set the new data connection to the layer's CIM featureTable
26 lyrCIM.featureTable.dataConnection = dc
27
28 # Set the layer's CIM definition
29 l.setDefinition(lyrCIM)
```

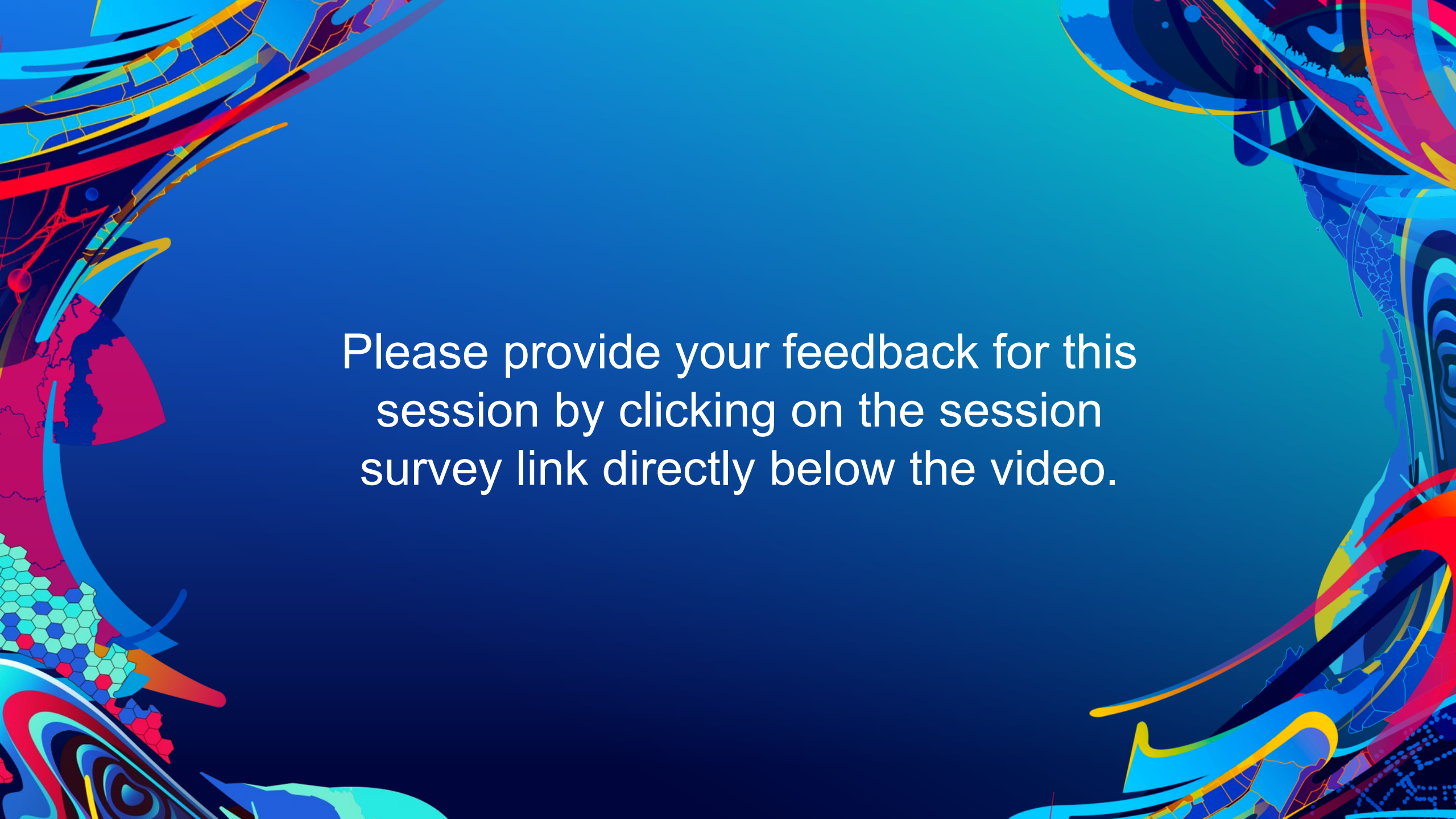

CIM: Updating map service URLs

```
1 # Reference project, map and layer
2 p = arcpy.mp.ArcGISProject(r"C:\Projects\YosemiteNP\Yosemite.aprx")
3 m = p.listMaps('Map Service')[0]
4 l = m.listLayers('World Light Gray Canvas Base')[0]
5
6 # Get the layer's CIM definition
7 lyrCIM = l.getDefinition('V2')
8
9 # Update the layer's URL and name
10 lyrCIM.serviceConnection.url = "https://services.arcgisonline.com/arcgis/rest/services/Canvas/World_Dark_Gray_Base/MapServer"
11 lyrCIM.name = "World Dark Gray Canvas Base"
12
13 l.setDefinition(lyrCIM)
```



esri®

THE
SCIENCE
OF
WHERE®

The background is a vibrant, abstract composition. It features a central area of solid blue and teal. The left and right sides are framed by complex, colorful patterns. On the left, there are swirling shapes in red, yellow, and blue, along with a section of a hexagonal grid in shades of green and blue. On the right, there are more swirling patterns in blue, red, and yellow, with some areas resembling a stylized face or mask. The overall effect is dynamic and visually stimulating.

Please provide your feedback for this session by clicking on the session survey link directly below the video.

Q & A



Jeff Barrette
Esri – Olympia, WA



Jeff Moulds
Esri – Redlands, CA