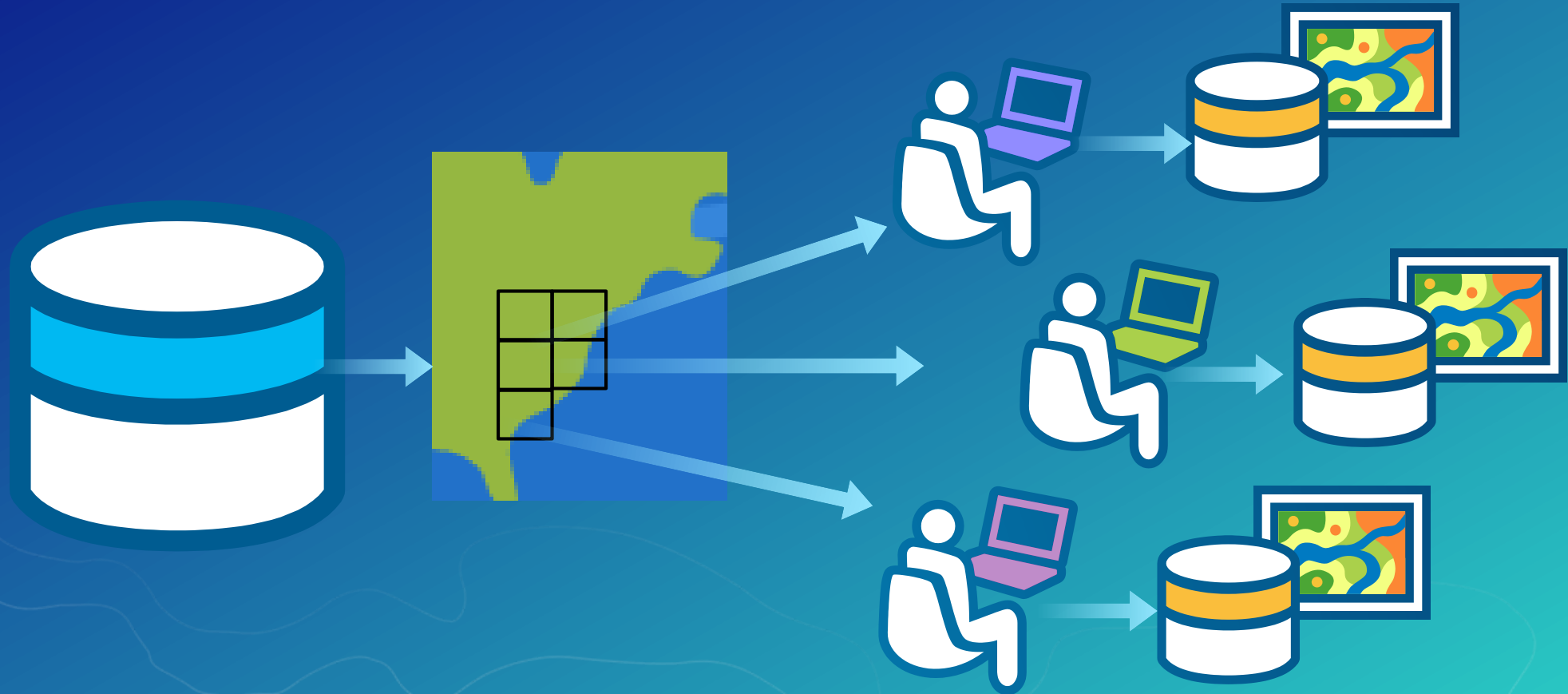


Esri Production Mapping: Map Automation & Advanced Cartography

MADHURA PHATERPEKAR

JOE SHEFFIELD

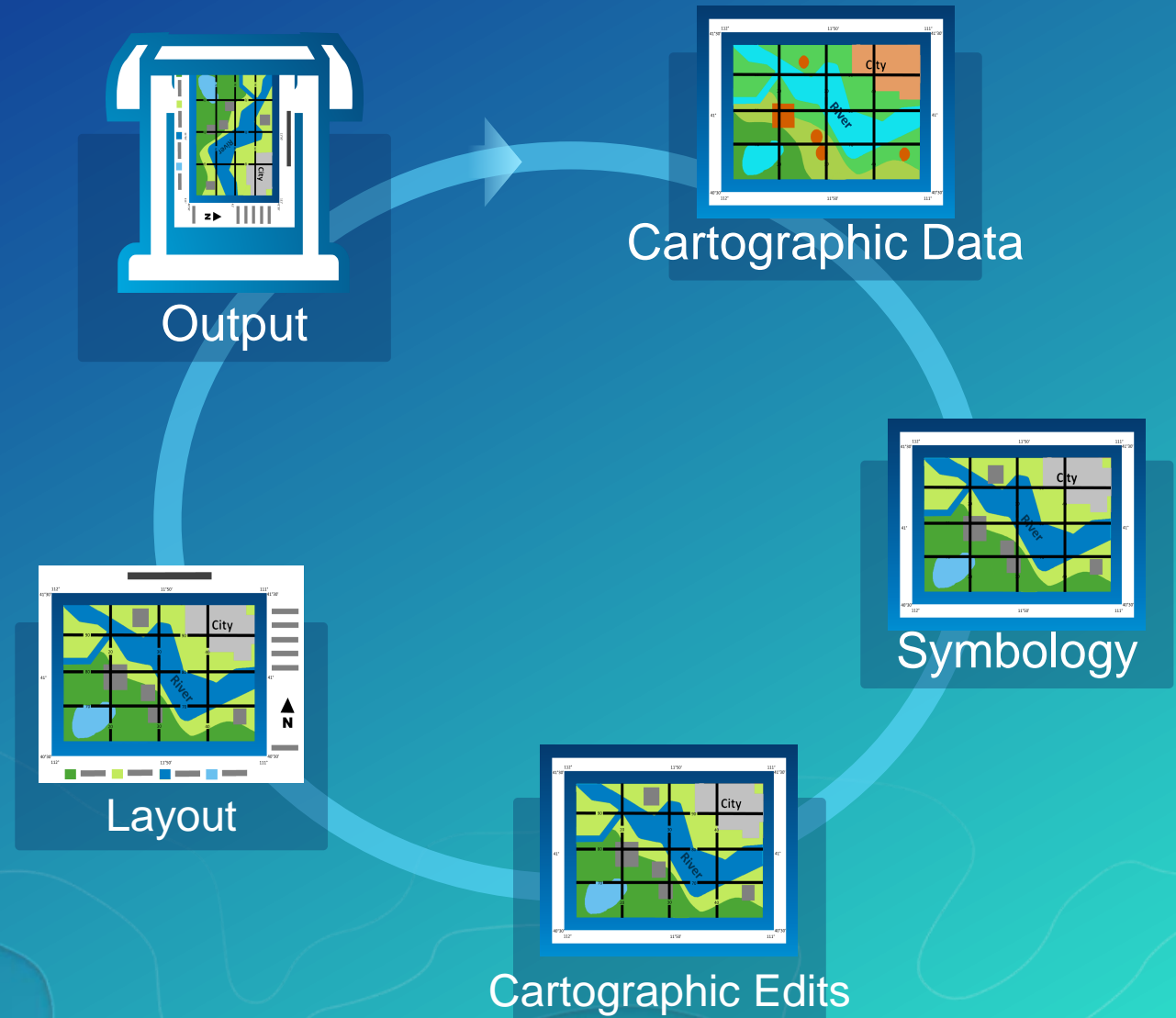
Traditional Cartography



What you really want

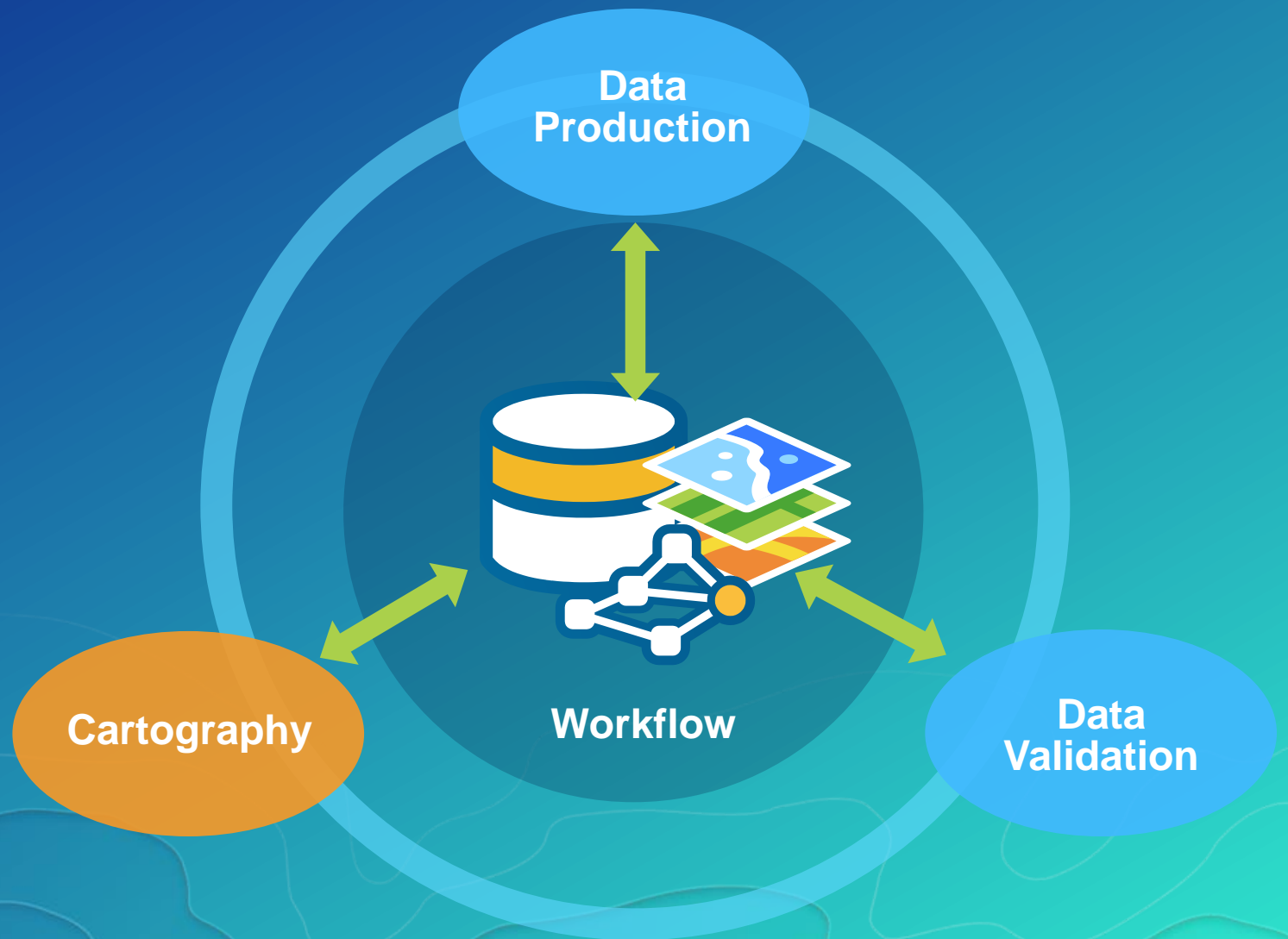


Cartographic Workflow

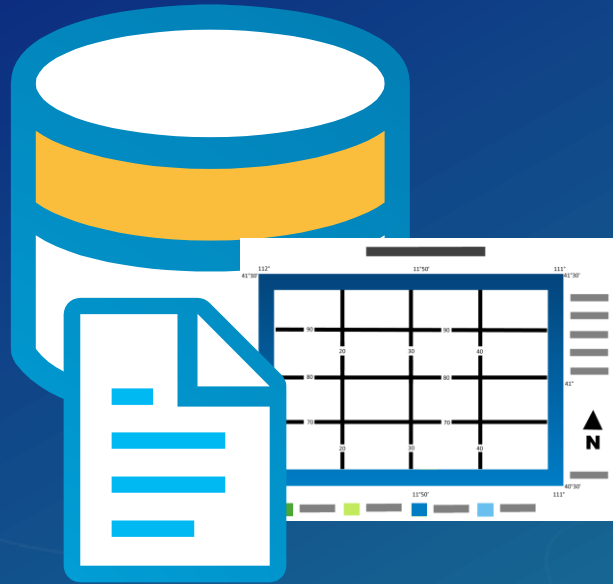


Esri Production Mapping

A collection of ArcMap Extensions

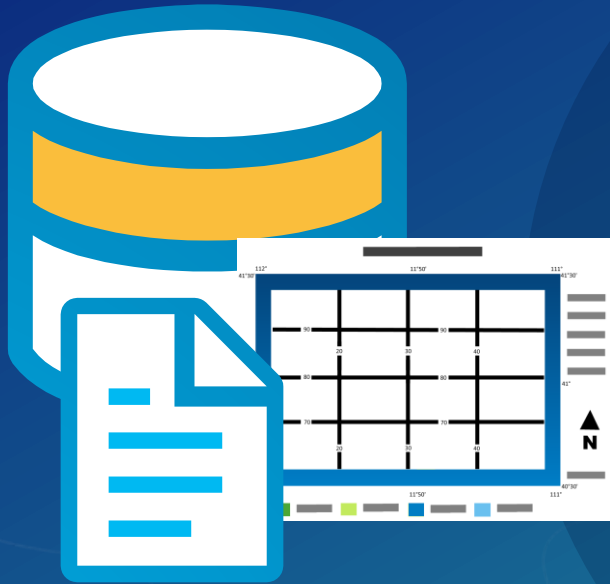


Map Automation



Production Mapping
Cartographic Rules

Map Automation



Production Mapping
Cartographic Rules



Geoprocessing
& Python

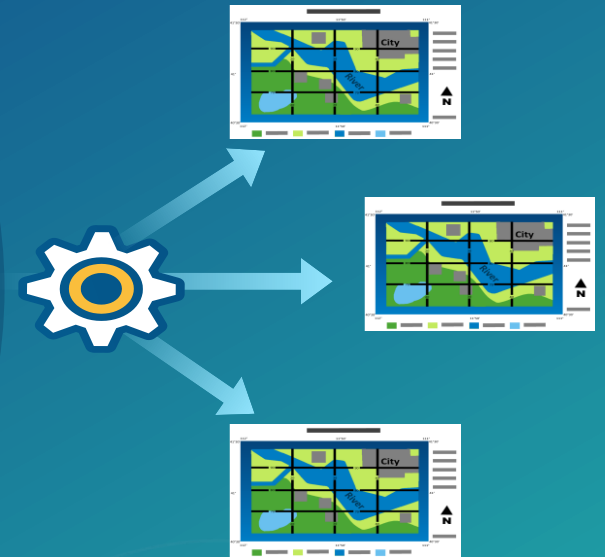
Map Automation



Production Mapping
Cartographic Rules



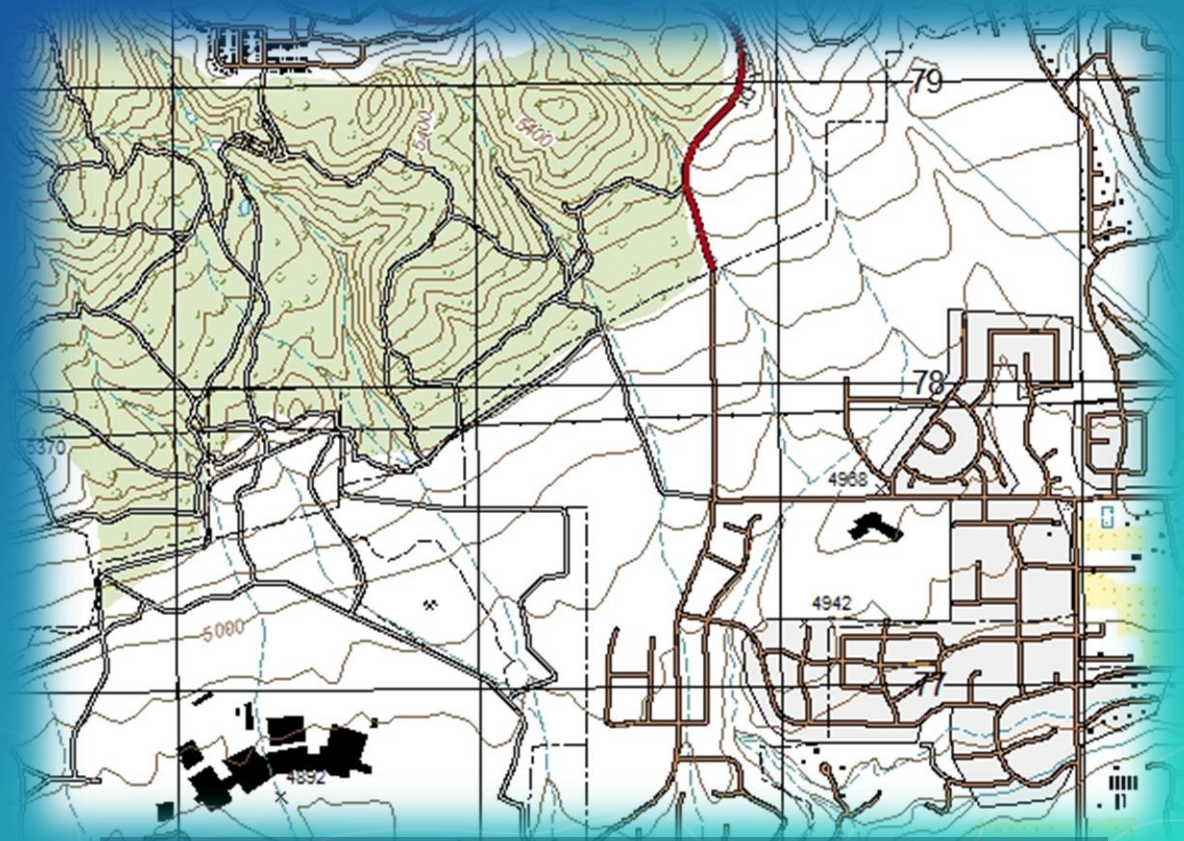
Geoprocessing
& Python



Standard
Output

Civilian Topographic Model (CTM)

- Based on the National System for Geospatial Intelligence Feature Data dictionary (NFDD)
- Includes:
 - Database schema
 - Editing Rules
 - Quality Control Rules
 - Cartographic templates for 25K
 - Cartographic templates for 50K
 - 50K Generalization Models
 - Workflow Manager Workflows
 - Distributed Generalization Workflows

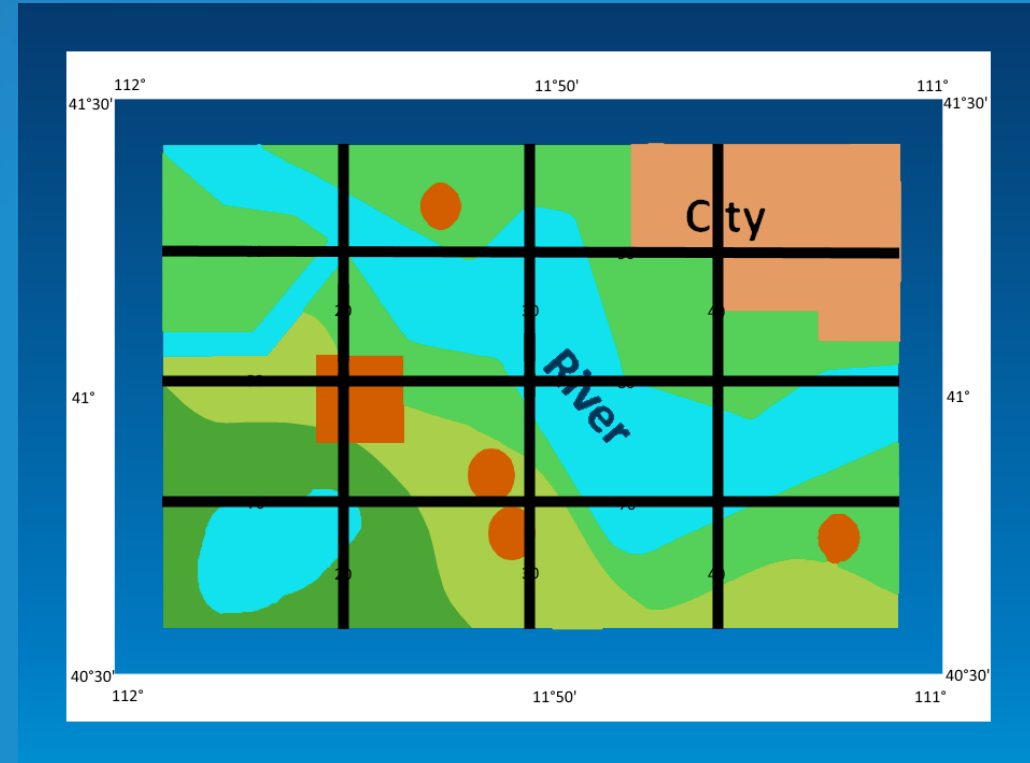
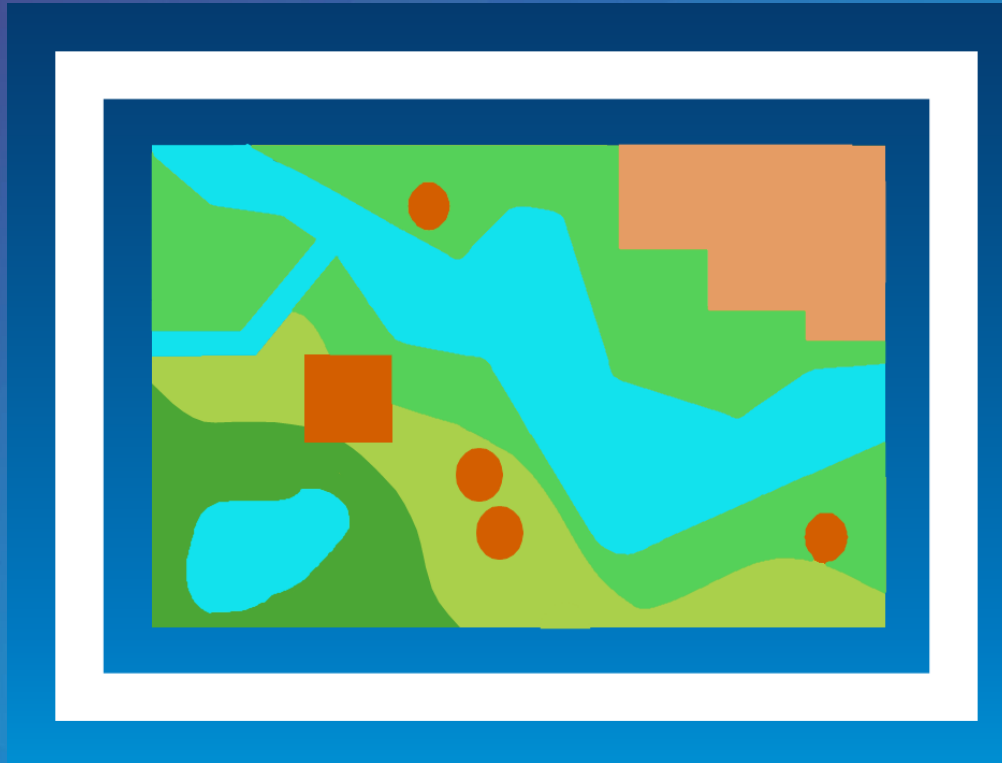


<https://github.com/esri/ctm>

Cartographic Data



Cartographic Data

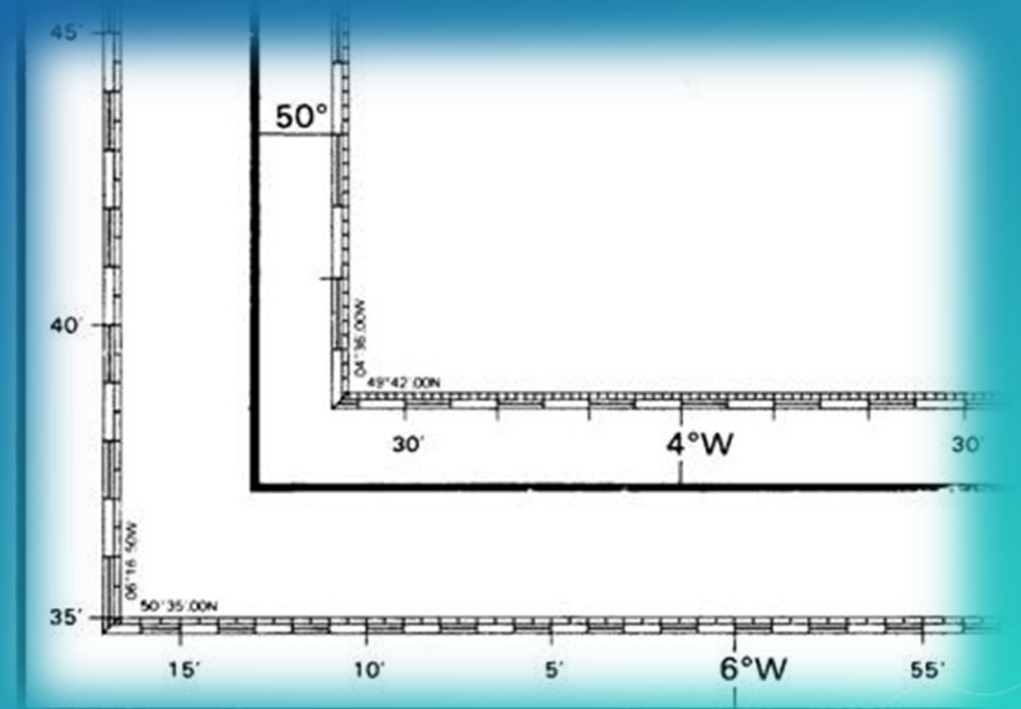
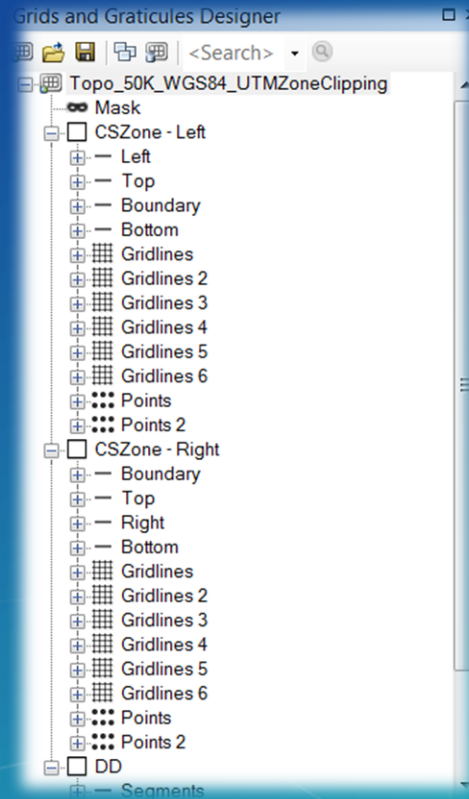


- Data and text used to enrich cartographic products
- Create data appropriate for your cartographic scale

Grids and Graticules

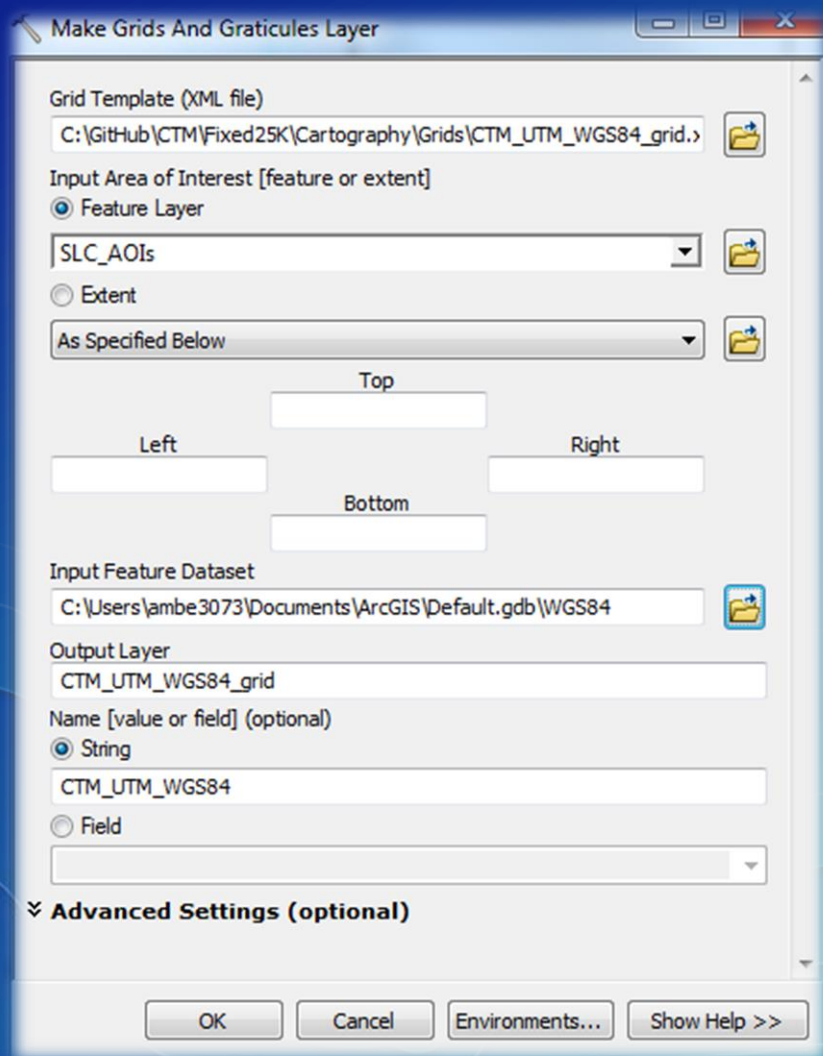
Design

- Feature based
- Fine Grain Custom Design
- Geographically aware
 - Scale
 - Coordinate system
 - Rotation



Grids and Graticules

Applying



```
#Define grid object
grid = arcpyproduction.mapping.Grid(grid_xml)

#Uses the appropriate XML for to create the grid
arcpy.AddMessage("Creating the Grid...")
output_layer = map_name + '_' + grid.type
grid_result = arcpy.MakeGridsAndGraticulesLayer_cartography(grid_xml,
    aoi, gfd, output_layer, map_name)
grid_layer = grid_result.getOutput(0)

# Add the grid layer to the top of the map
arcpy.mapping.AddLayer(data_frame, grid_layer, "TOP")
arcpy.AddMessage("Grid Layer added to map...")

# Updates the data frame properties base on the grid
final_mxd.activeView = 'PAGE_LAYOUT'
grid.updateDataFrameProperties(data_frame, aoi)
```


Generalization

- Resolves the appearance of feature geometry at smaller scales



1: 25000



1: 50000

Feature Generalization



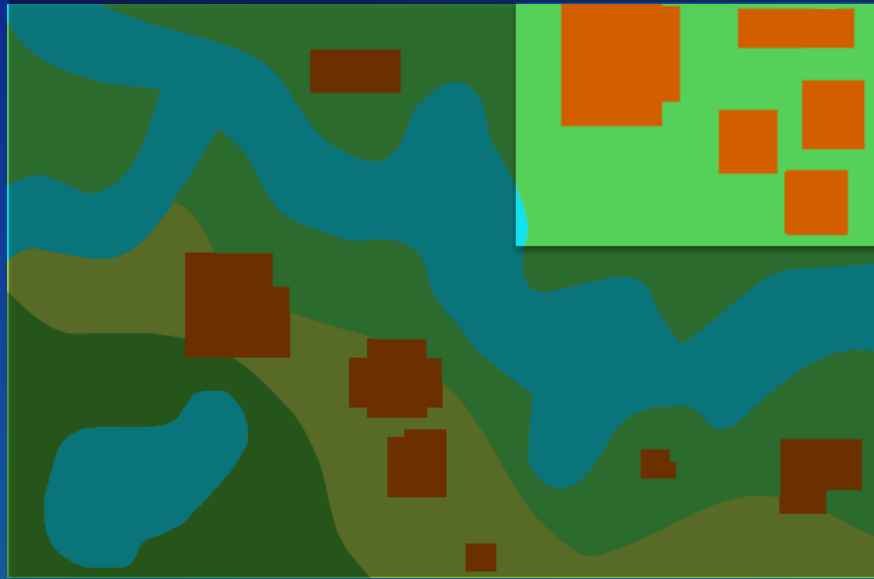
1: 25000



1: 50000

- Features assessed individually without regard to symbology or spatial relationships

Contextual Generalization



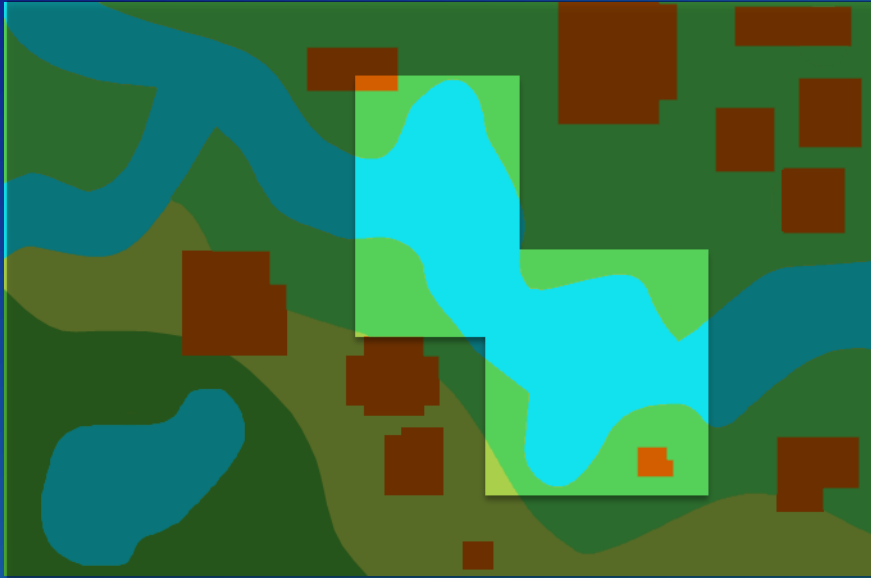
1: 25000



1: 50000

- Features are assessed collectively
 - Maintain pattern, density, and spatial relationships

Contextual Generalization



1: 25000



1: 50000

- Features from multiple layers assessed simultaneously
 - Maintain pattern, density, and spatial relationships

Generalization Tools

Production Mapping Tools toolbox

- + Cartographic Data
- + Cartographic Editing
- + Conversion
- + Distributed Geodatabase
- Generalization
 - Aggregate Polygons
 - Convert Polygons
 - Delete Dangles
 - Delete Polygons And Extend Lines
 - Extend Polygon Sides
 - Fill Gaps
 - Generalize Shared Features
 - Increase Line Length
 - Increase Polygon Area
 - Modify Underlying Polygon
 - Thin Hydro Features
 - Thin Spot Heights
- + Product Library
- + Surround Elements

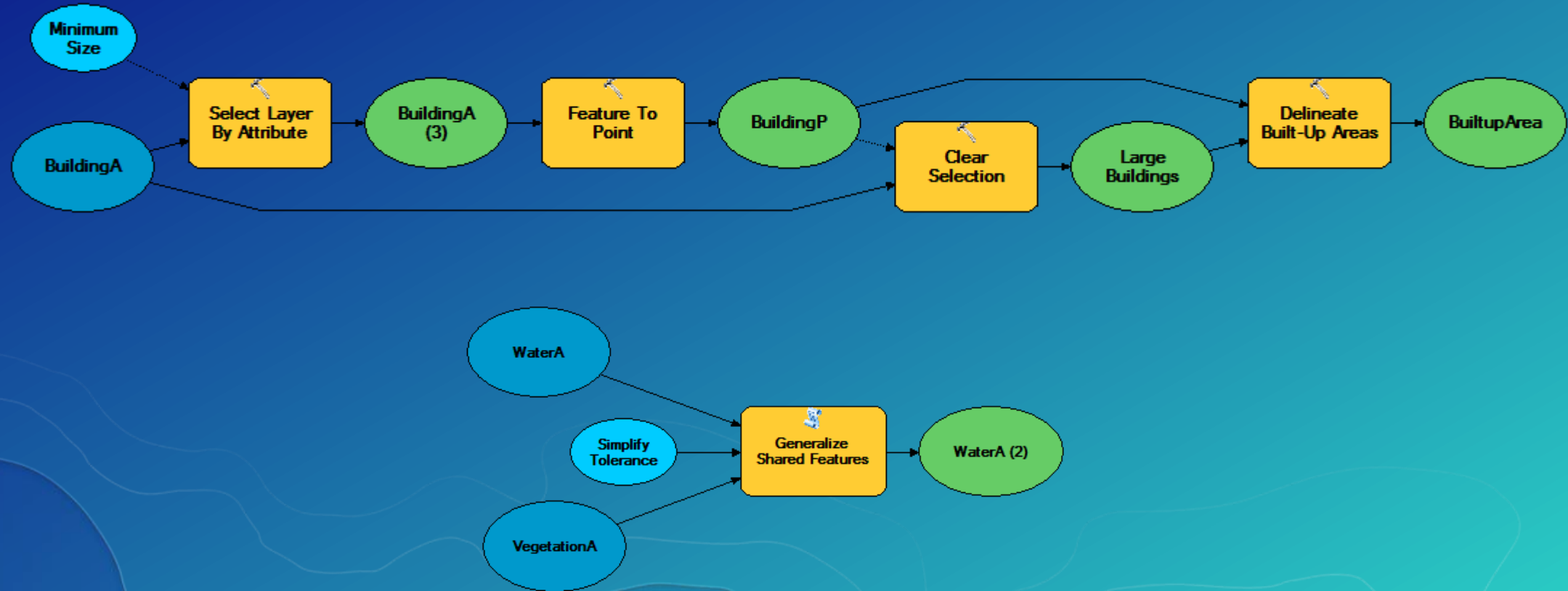
Cartography Tools toolbox

- + Annotation
- + Cartographic Refinement
- + Data Driven Pages
- Generalization
 - Aggregate Points
 - Aggregate Polygons
 - Collapse Dual Lines To Centerline
 - Collapse Road Detail
 - Create Cartographic Partitions
 - Delineate Built-Up Areas
 - Merge Divided Roads
 - Simplify Building
 - Simplify Line
 - Simplify Polygon
 - Smooth Line
 - Smooth Polygon
 - Thin Road Network
- Graphic Conflicts
 - Detect Graphic Conflict
 - Propagate Displacement
 - Resolve Building Conflicts
 - Resolve Road Conflicts
- + Grids and Graticules
- + Masking Tools
- + Representation Management

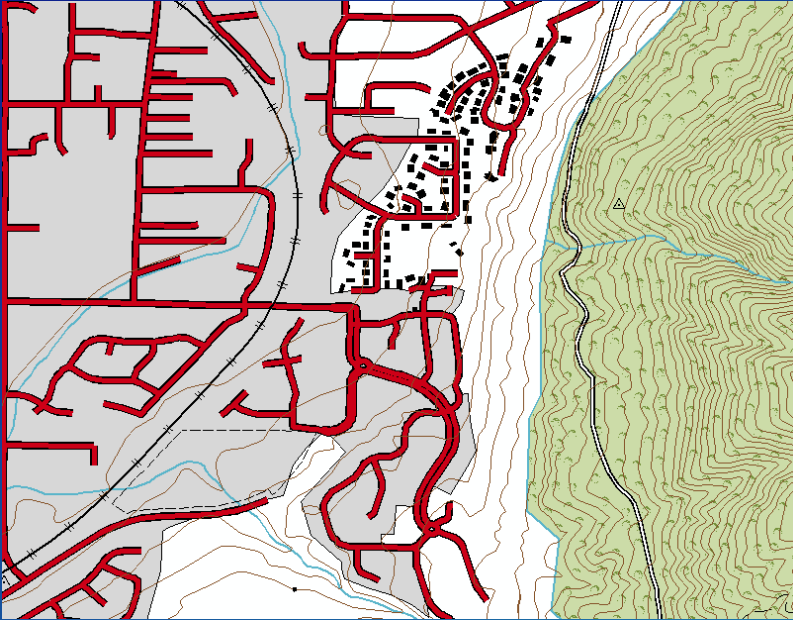
Data Management Tools toolbox

- + Archiving
- + Attachments
- + Data Comparison
- + Distributed Geodatabase
- + Domains
- + Feature Class
- Features
 - Add Geometry Attributes
 - Add XY Coordinates
 - Adjust 3D Z
 - Bearing Distance To Line
 - Check Geometry
 - Copy Features
 - Delete Features
 - Dice
 - Feature Envelope To Polygon
 - Feature To Line
 - Feature To Point
 - Feature To Polygon
 - Feature Vertices To Points
 - Geodetic Densify
 - Minimum Bounding Geometry
 - Multipart To Singlepart
 - Points To Line
 - Polygon To Line
 - Repair Geometry
 - Split Line at Point
 - Split Line At Vertices
 - Table To Ellipse
 - Unsplit Line
 - XY To Line

Find the tool and add to a model



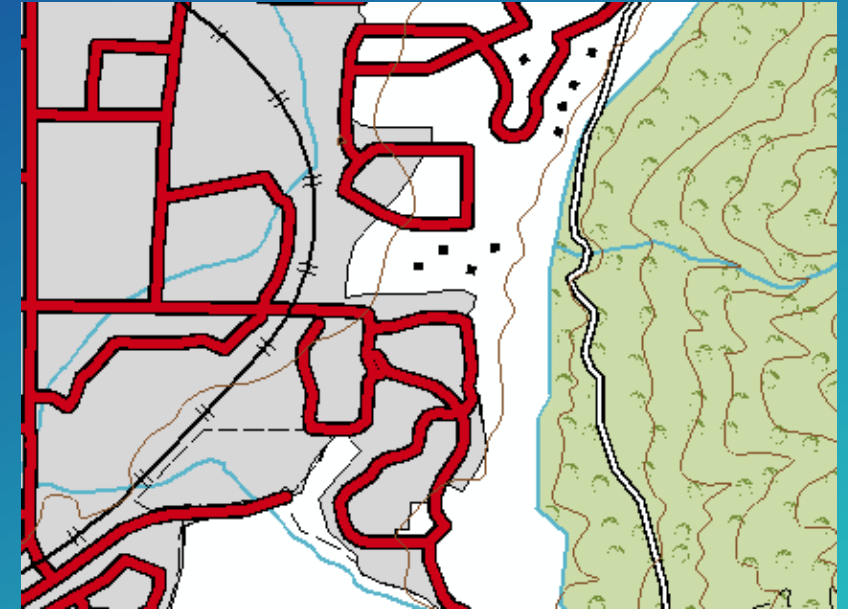
Run the models



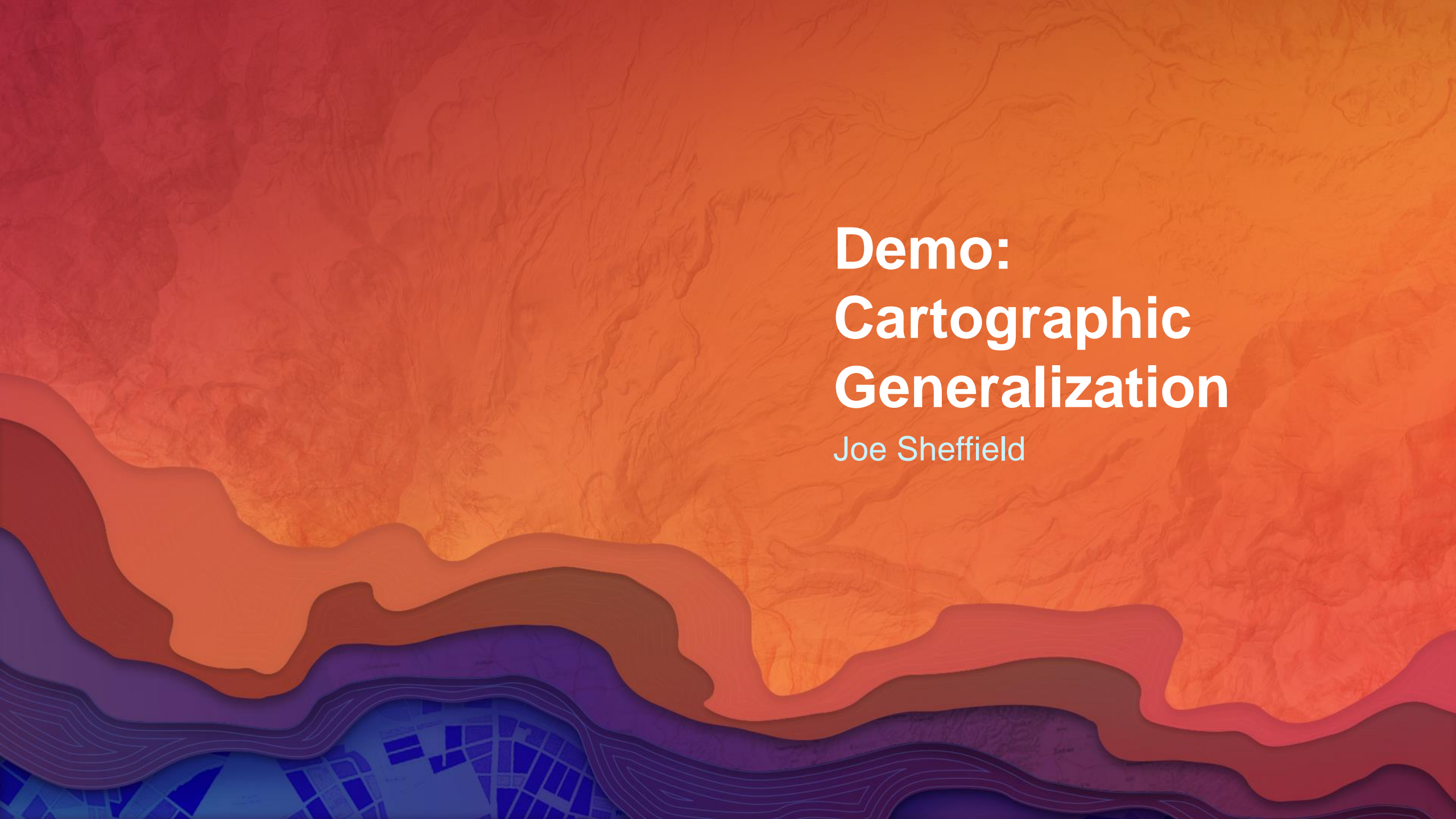
1: 25000



- Fixed50K
- Generalization
- CTM_50K_Generalization.tbx
 - Generalization Submodels
 - Prepare Models
 - 0 Prepare Data for Generalization
 - 1 Transportation
 - 2 Buildings
 - 3 Hydrography
 - 4 Land Cover
 - 5 Elevation
 - 6 Apply Symbology
 - 7 Resolve Line Conflicts
 - 8 Resolve Structure Conflicts



1: 50000

The background is a stylized map. The upper portion is a warm-toned topographic map with contour lines in shades of orange and red. The lower portion features a blue-toned urban map with a grid street pattern, overlaid with wavy, layered shapes in various shades of blue and purple, suggesting water or a specific map projection effect.

Demo: Cartographic Generalization

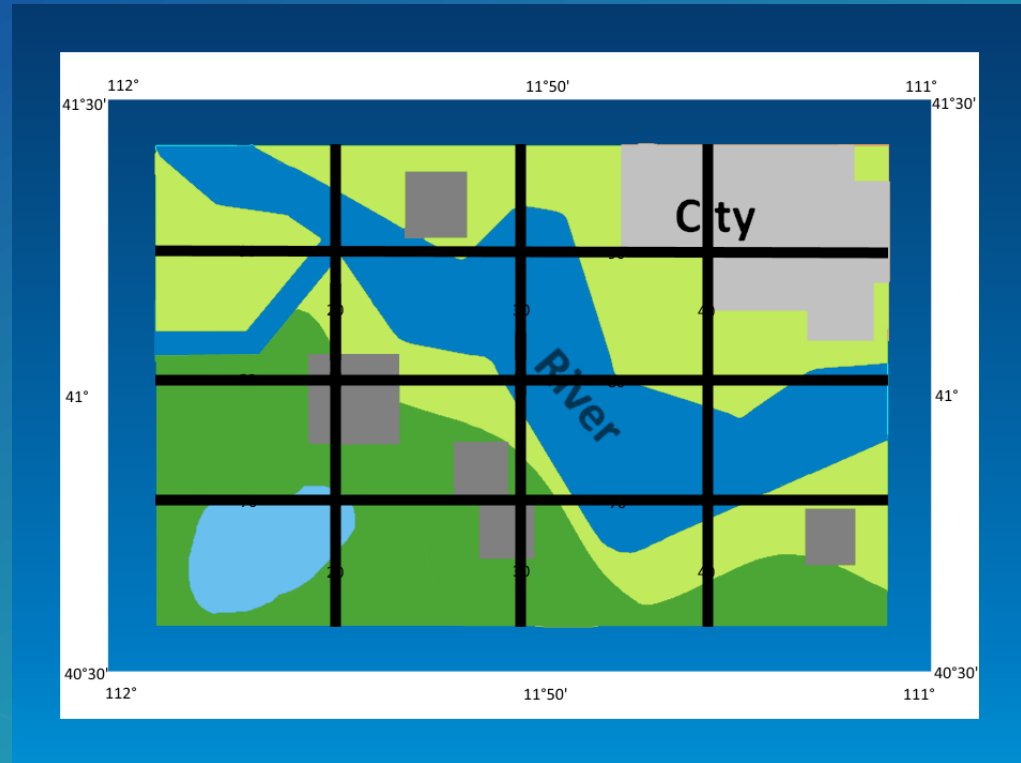
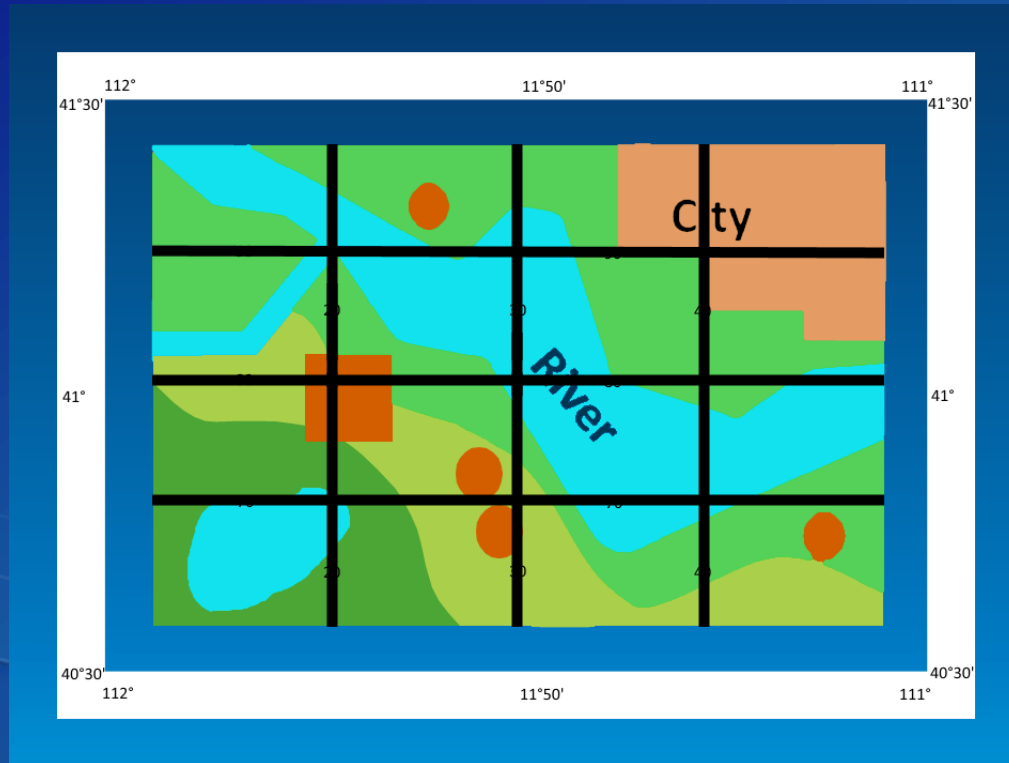
Joe Sheffield

Symbology

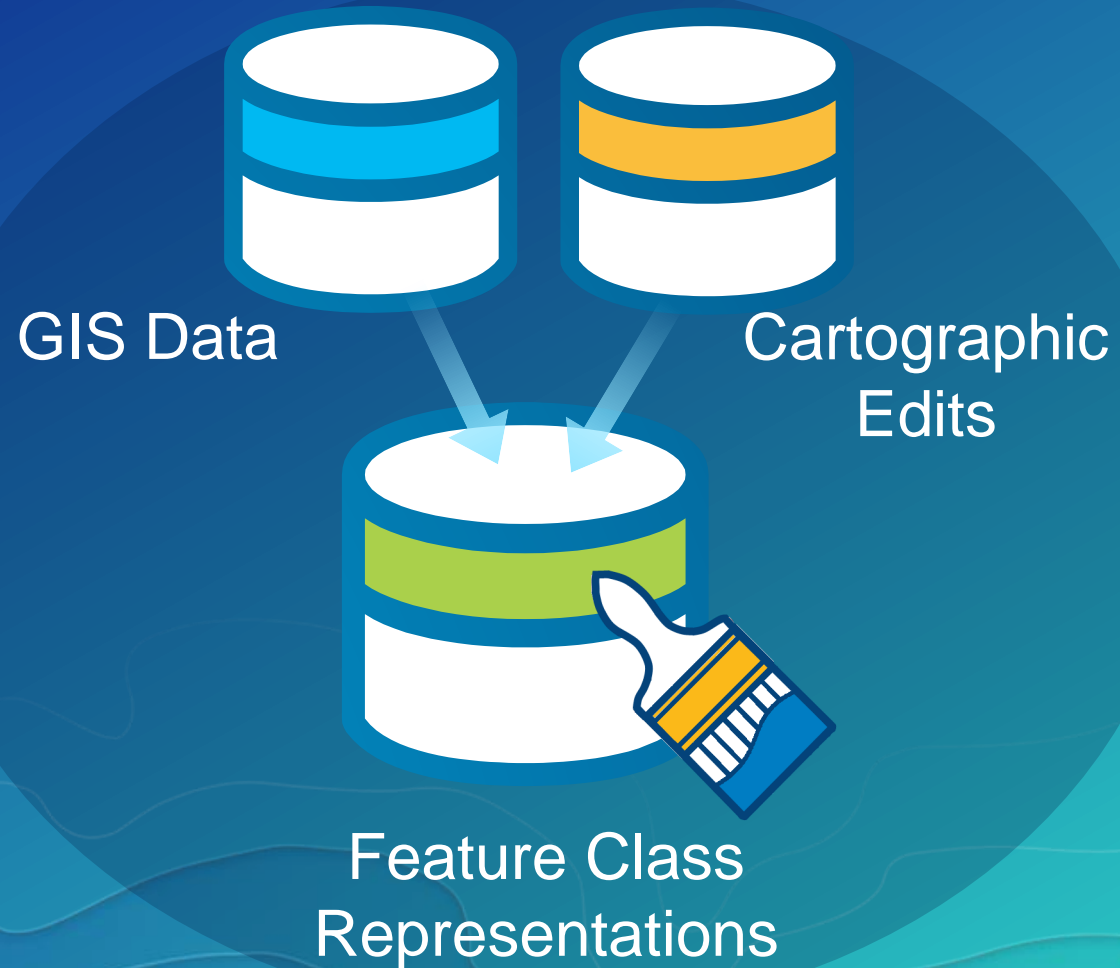


Symbology

Intuitively displaying information



Feature Class Representations



Visual Specifications

Defining

- Create Symbolology
- Know Your Rules
- Define Specifications

1L015 - BUILDING
POINT FEATURE... SYMBOL#1L015P011

ATTRIBUTES:

HWT - HOUSE OF WORSHIP TYPE
000 - UNKNOWN
002 - CATHEDRAL
003 - CHAPEL
004 - CHURCH
NAM - NAME CATEGORY
ANY

PARAMETERS:

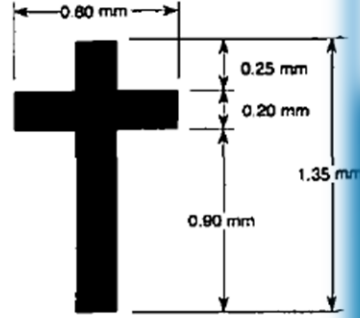
Origin: Center of square

FILL: SQUARE
Lineweight: 0.630mm
Length: 0.630mm
Color: BLACK (#58600)

LABEL: NAM
Type Size: 6 point
Type Style: Upper/Lower Case, Condensed
Color: RED (#60862)

POSICUT
Posicut: #21
Color: BLACK (#58600)

POSICUT # 21



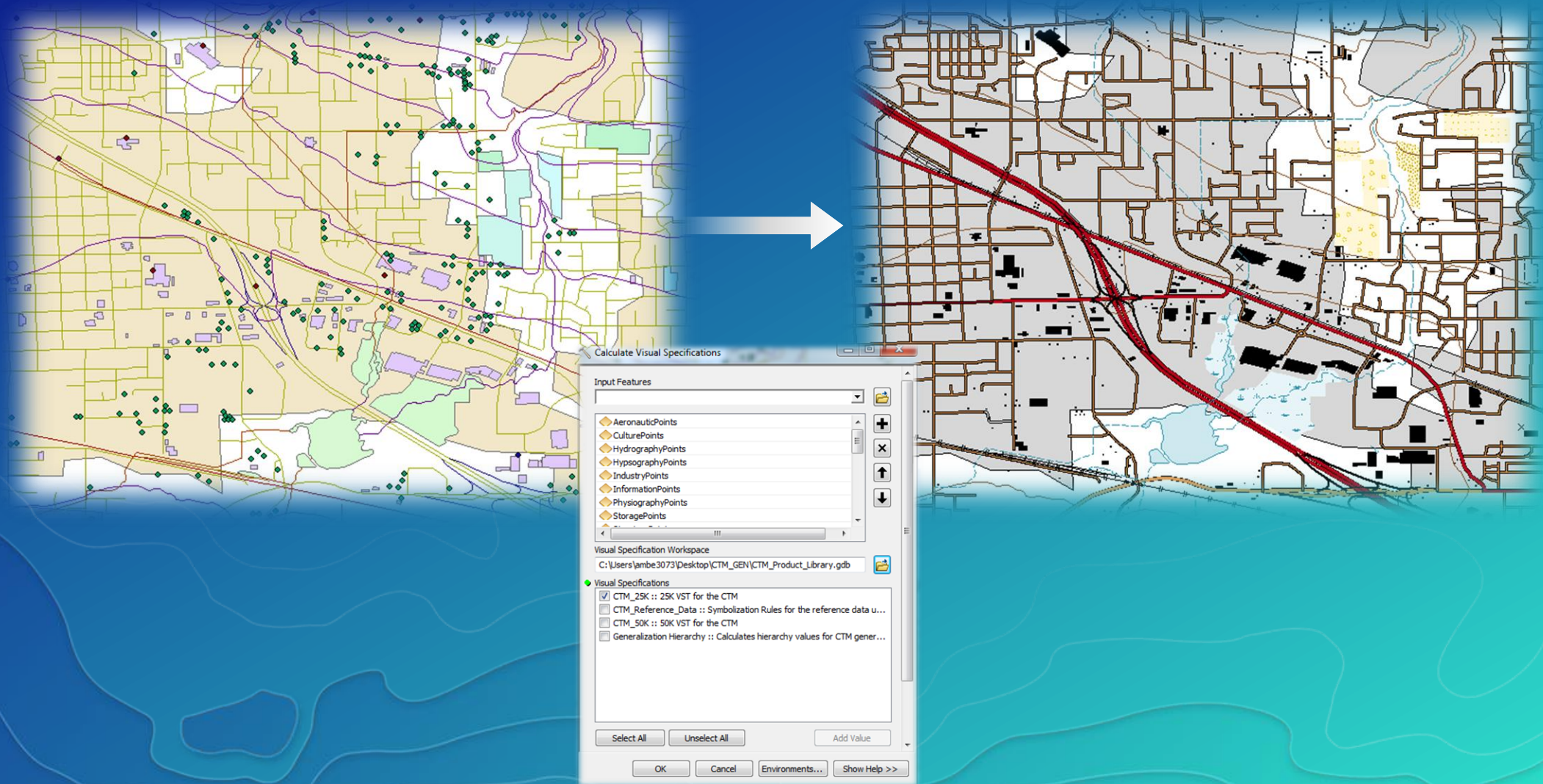
Cross lineweights : 0.20 mm
Origin : Base of cross

Rule Description	Dataset	SQL Statement	Preview
Tower	StructurePnt	SELECT <TargetTable.OID> FROM <TargetTable> WHERE F_CODE = 'AL241'	Structure: Tower (with Annotation) Pnt
Church	StructurePnt	SELECT <TargetTable.OID> FROM <TargetTable> WHERE HWT in (000, 002, 003, 004)	Structure: Building, HOW - Christian Pnt
Building/Shed	StructureSf	SELECT <TargetTable.OID> FROM <TargetTable> WHERE F_CODE = 'AL013' OR F_CODE = 'AL014'	Structure: Building

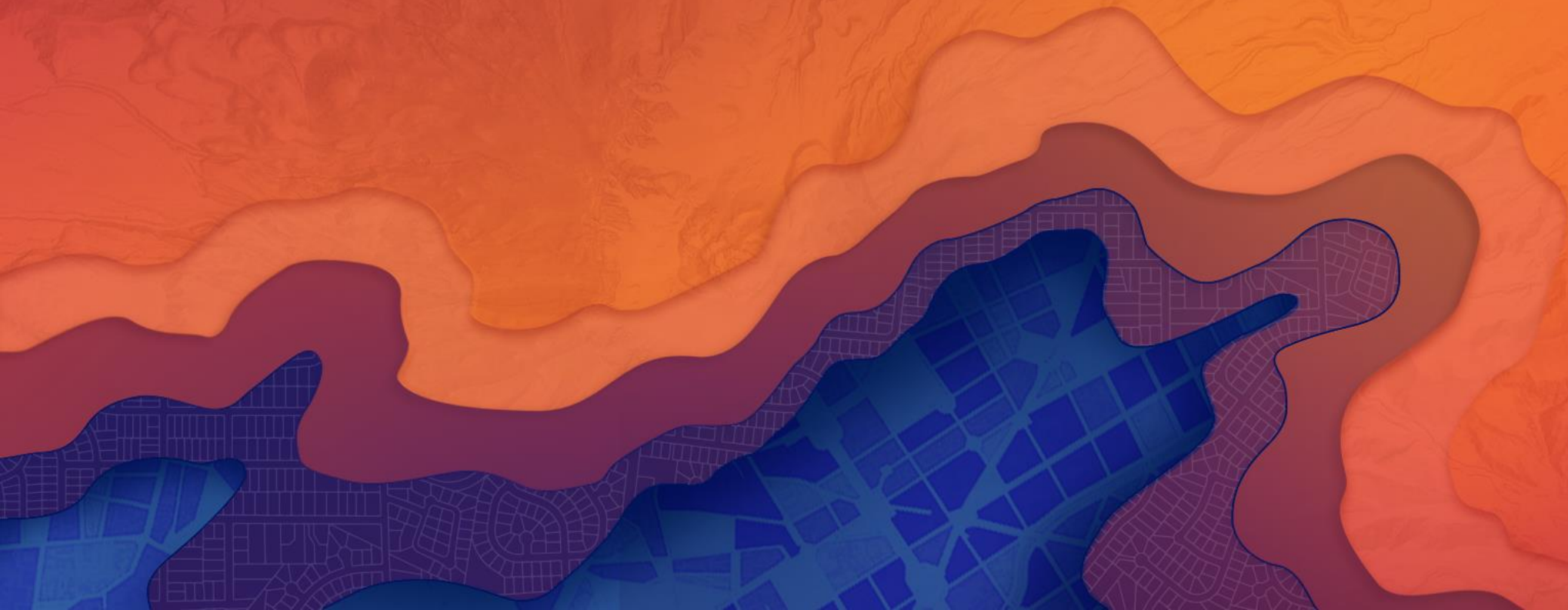
Options >> OK Cancel

Visual Specifications

Applying



Cartographic Edits

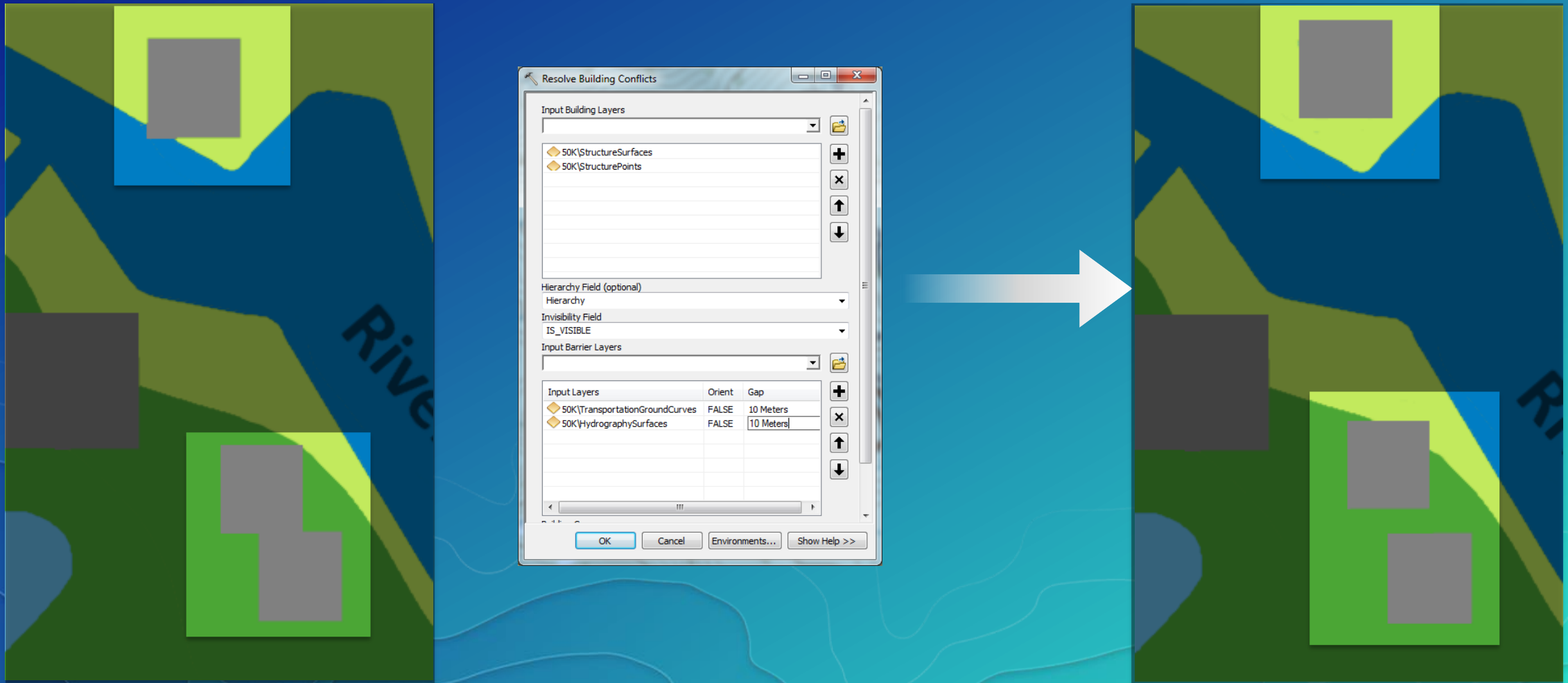


Feature Class Representations

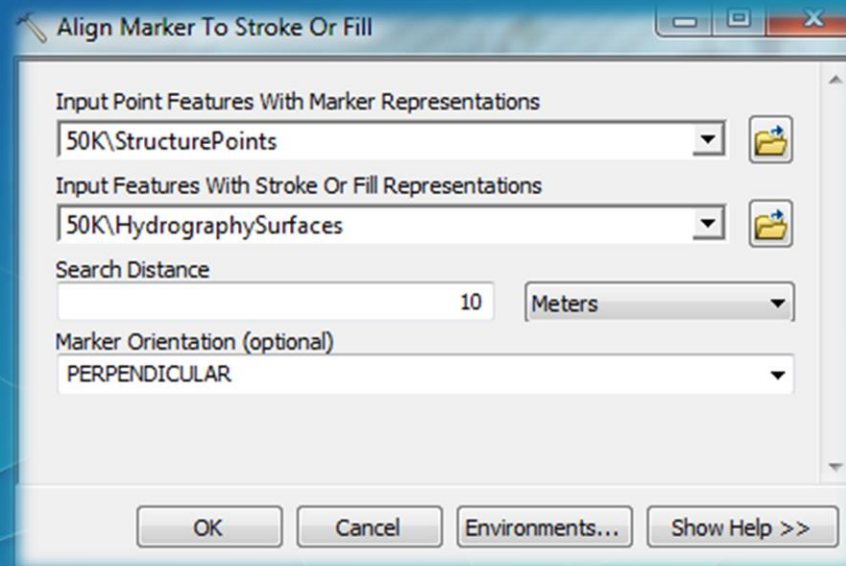


Conflict Resolution

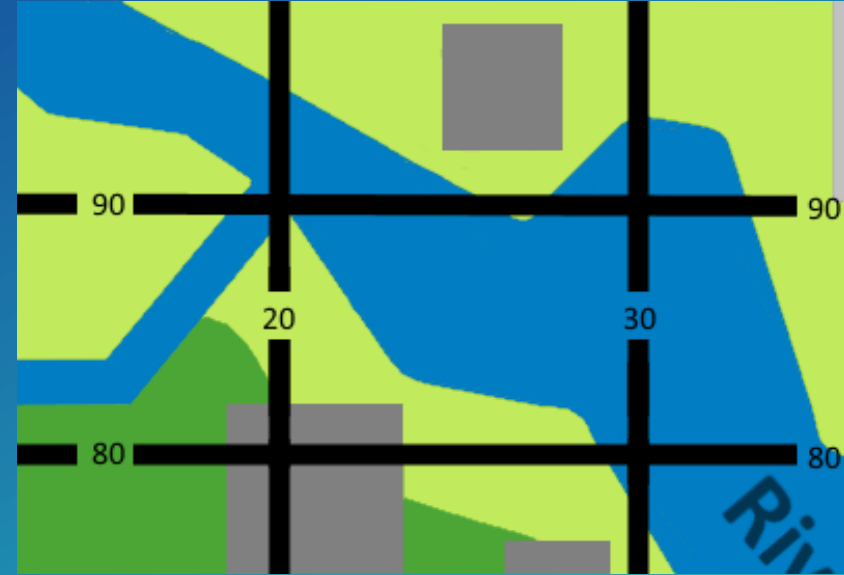
Managing placement of symbolized features



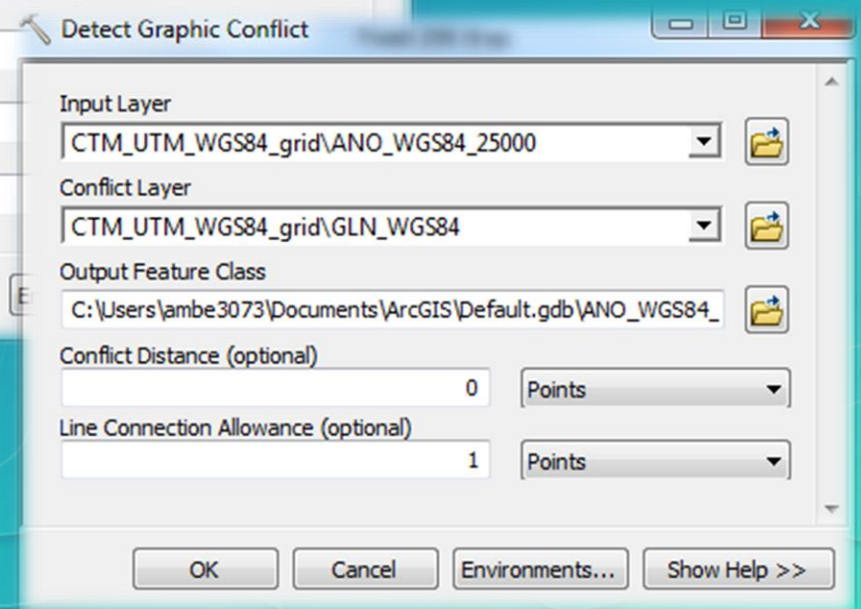
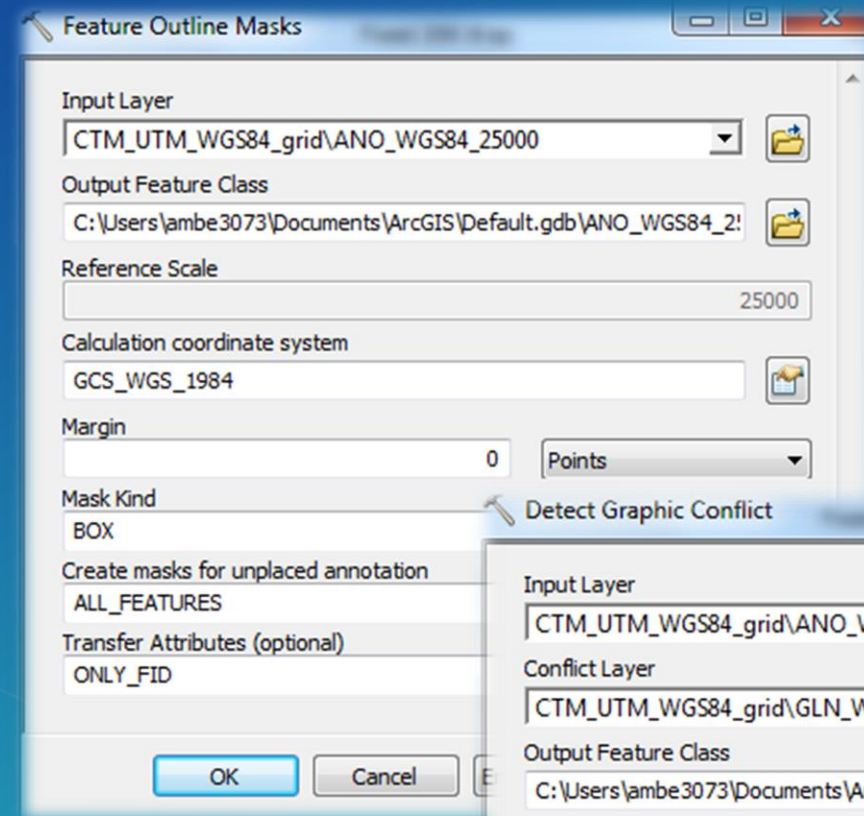
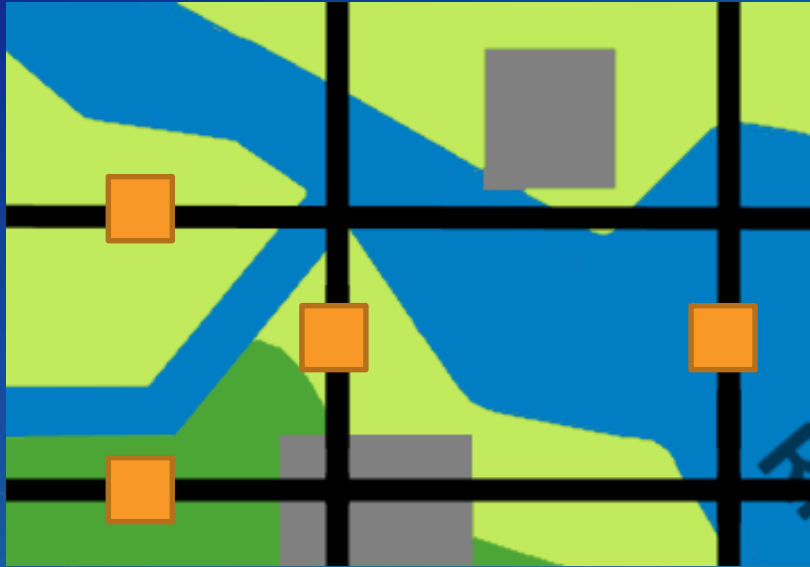
Cartographic Refinement



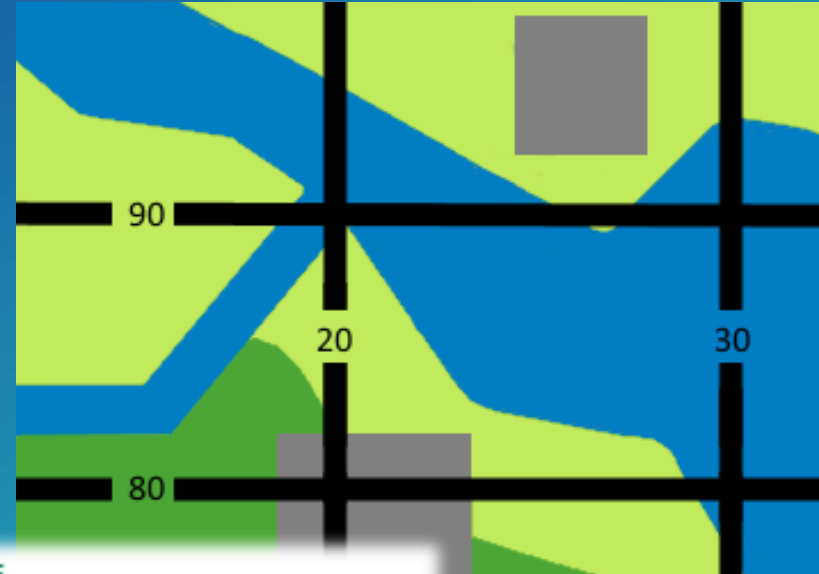
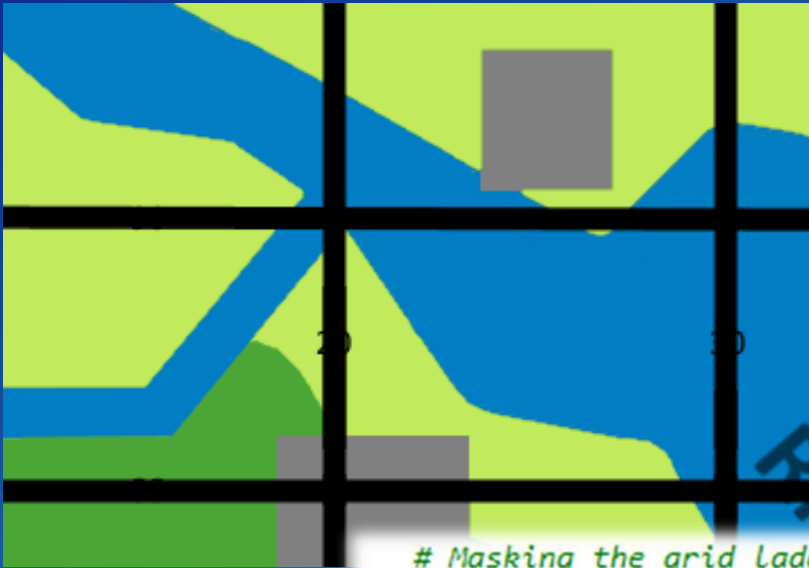
Masking



Masking



Masking



```
# Masking the grid ladder values and annotations
arcpy.AddMessage("getting output of masks.")
mask_layer = arcpy.mapping.Layer(masks.getOutput(0))
arcpy.mapping.AddLayer(data_frame, anno_mask_layer, 'BOTTOM')
anno_mask = arcpy.mapping.ListLayers(final_mxd, mask_layer.name,
                                     data_frame)[0]

arcpy.AddMessage("Annotation Mask layer added to the map...")
arcpyproduction.mapping.EnableLayerMasking(data_frame, 'true')
arcpyproduction.mapping.MaskLayer(data_frame, 'APPEND', anno_mask,
                                  gridline_layer)
arcpy.AddMessage("Masking applied to gridlines...")
```



Demo: Symbology & Cartographic Edits

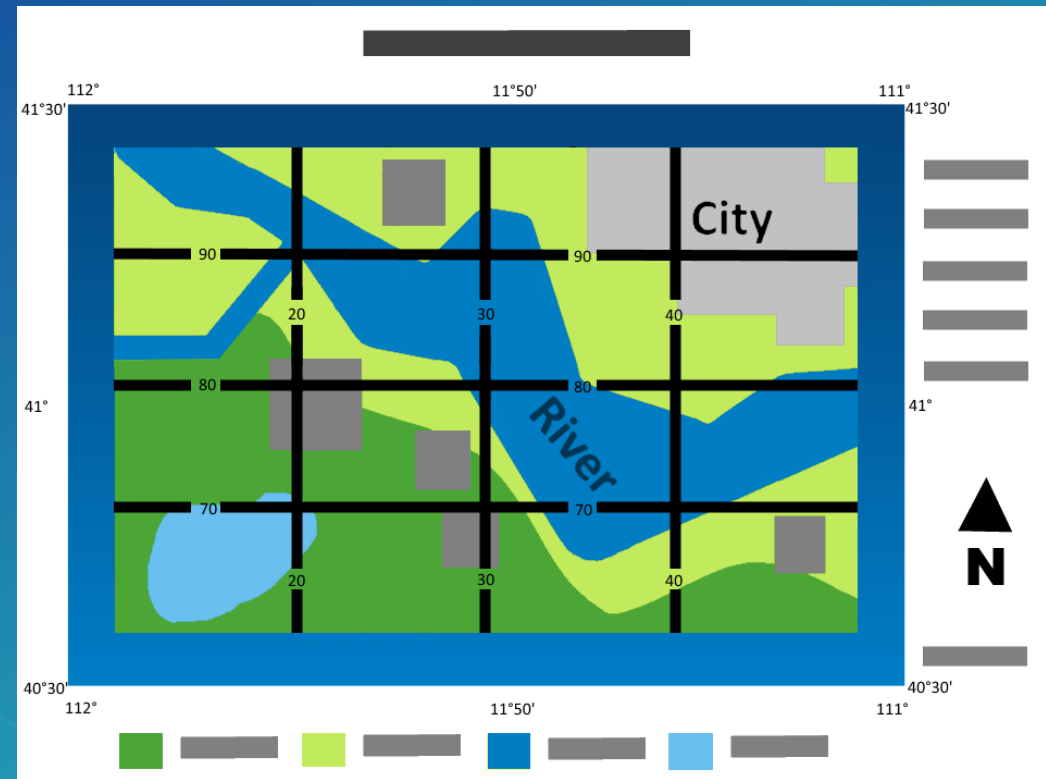
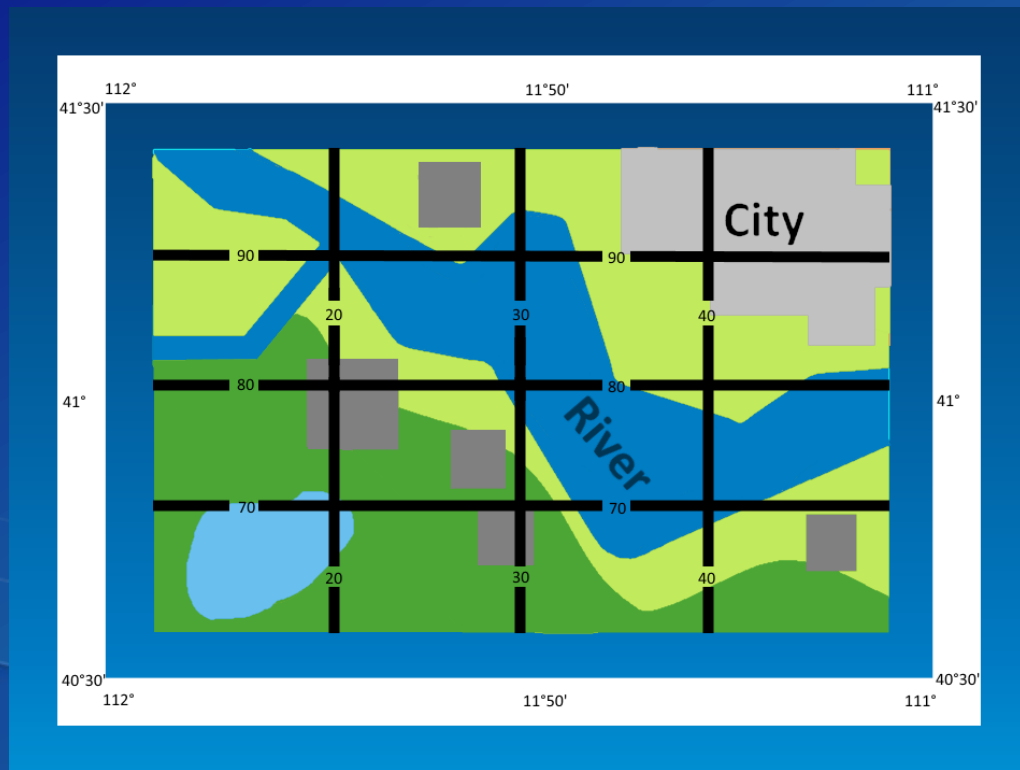
Joe Sheffield

Layout



Layout

Providing Context



Layout Elements

Types of Elements

- Static Elements



Producer
Information

Logo

Product
Name

Layout Elements

Types of Elements

- Static Elements
- Dynamic Elements
 - Dynamic Text
 - Dynamic Graphics
- Template Page



Sheet
Name

Angle
Of North

Map
Legend

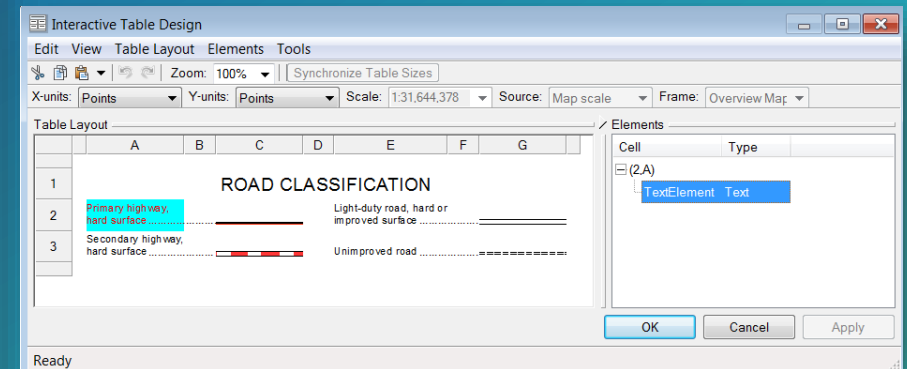
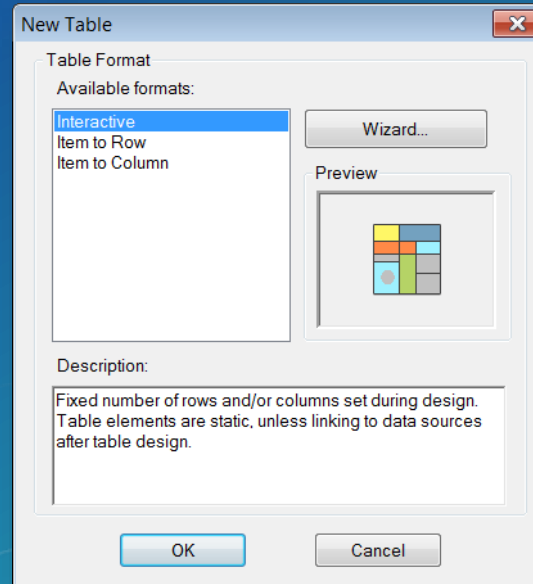
Layout Elements

Automating

- Static Elements
 - Dynamic Text
 - Python
 - Graphic Table Element
- Page Size

```
#Gets the list of layout elements
layout_elements = arcpy.mapping.ListLayoutElements(final_mxd)
for element in layout_element_list:

    #Update State Name text element
    if element.name == "State Name":
        element.text = state_name.upper()
```

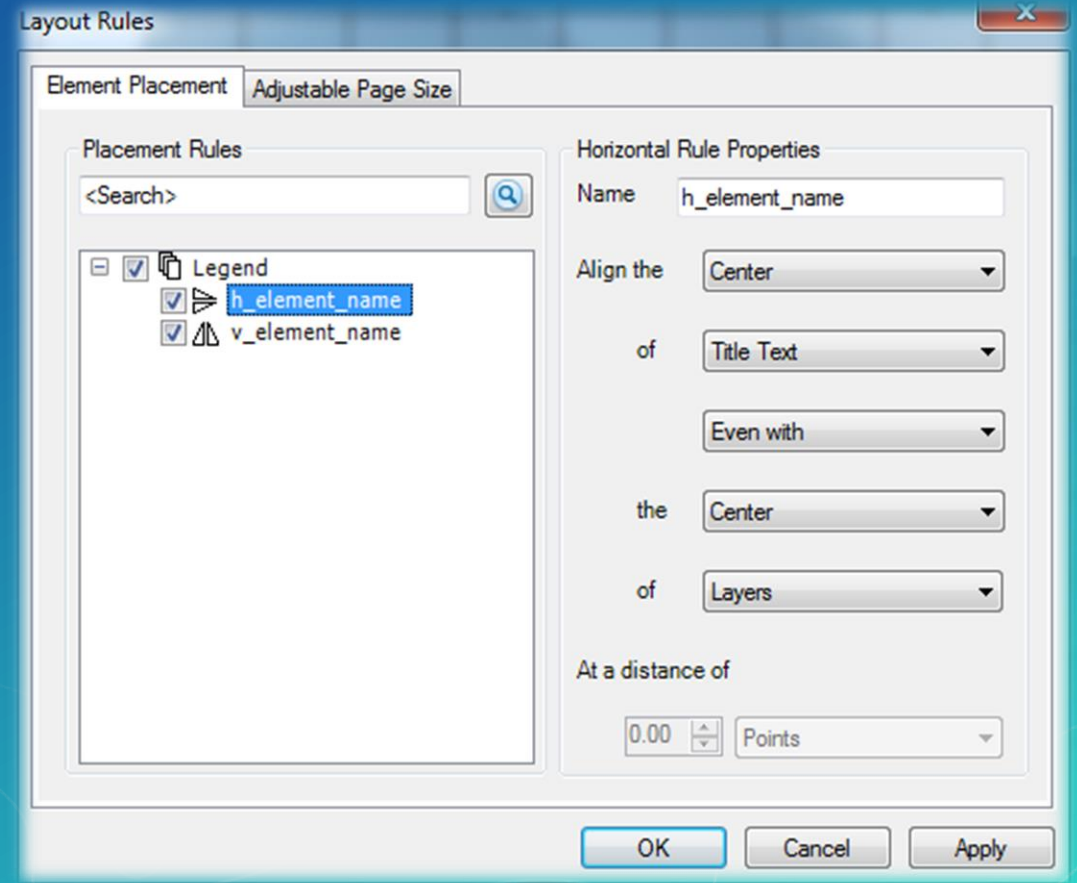


```
# Define variables
mxd = arcpy.mapping.MapDocument(r'C:\Project\Project.mxd')

# Set the page size
arcpyproduction.mapping.SetPageSize(mxd, "CUSTOM",
                                     custom_width=60, custom_height=40, page_units="CENTIMETERS")
```

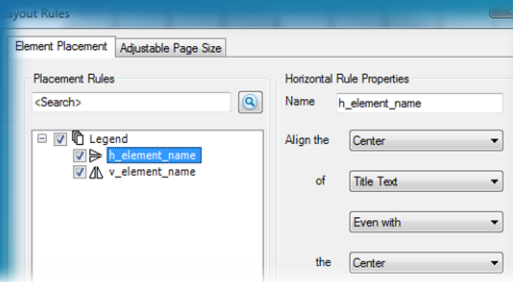
Layout Rules

Design



Layout Rules

Apply



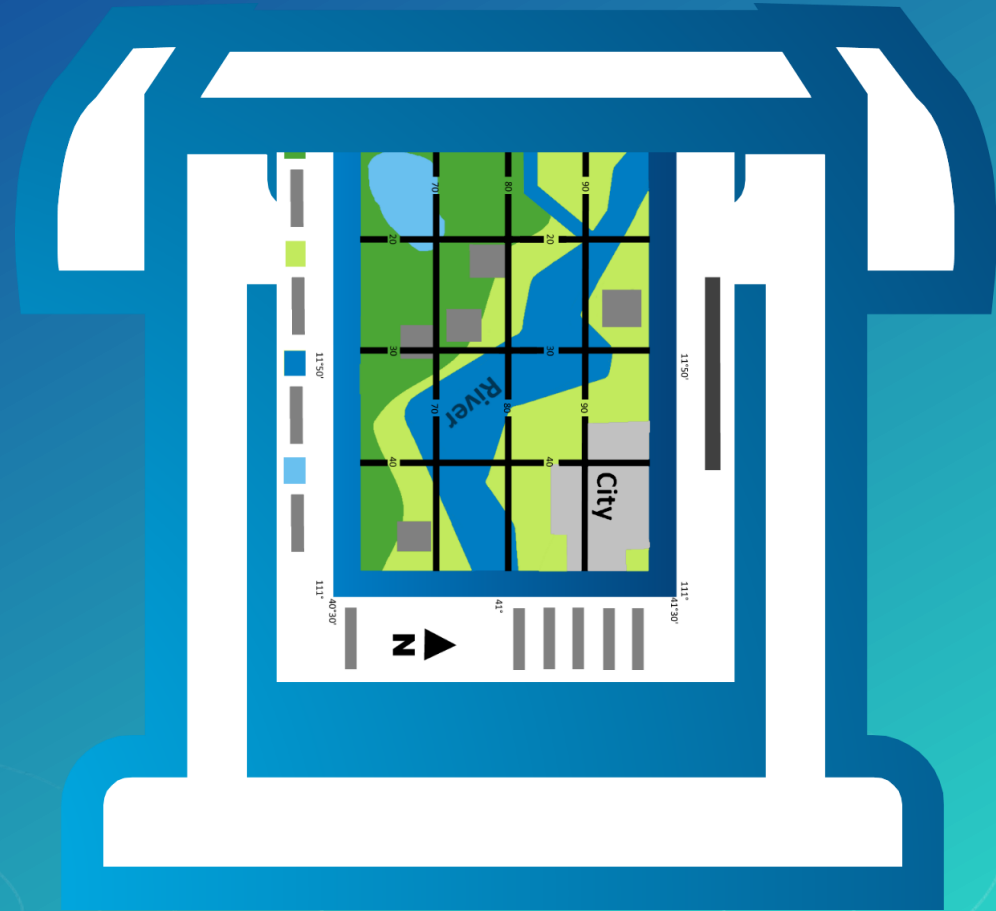
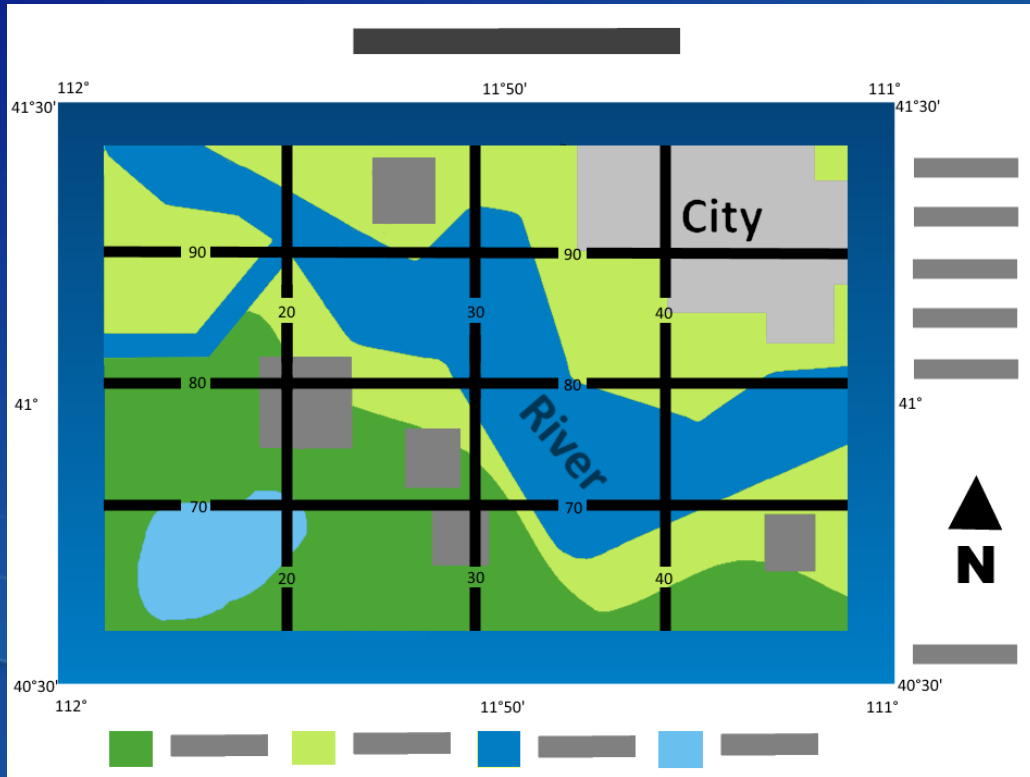
```
mxd = arcpy.mapping.MapDocument("CURRENT")
arcpyproduction.mapping.ApplyLayoutRules(mxd, layout_rules.xml)
```

Output Product



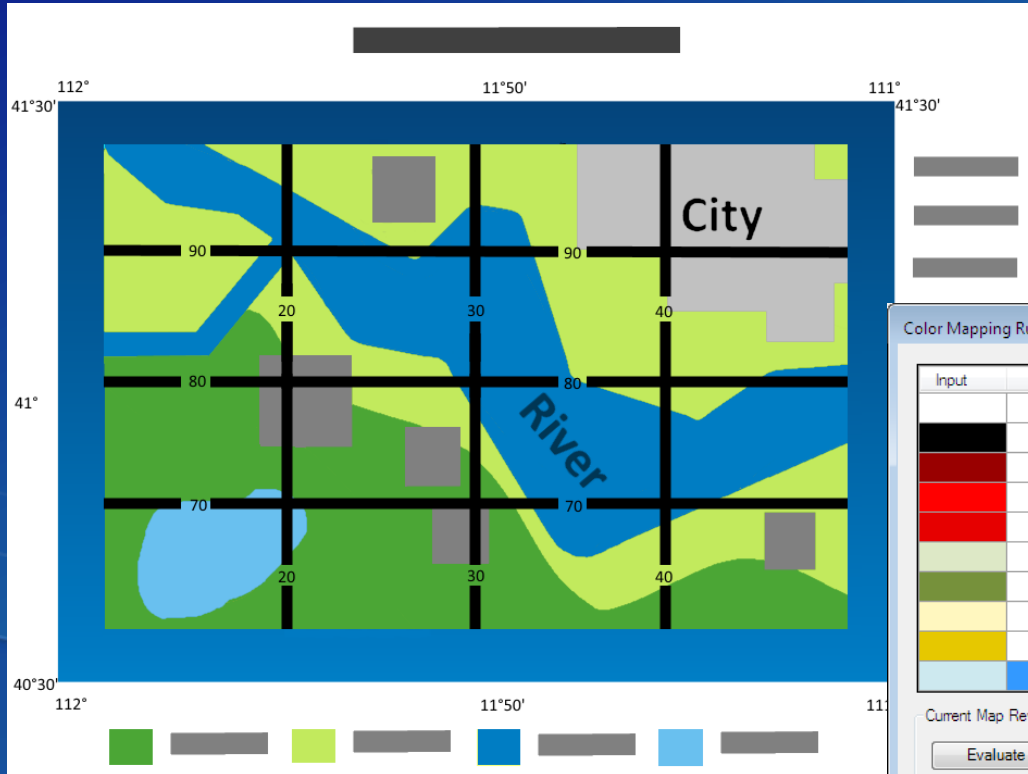
Product/ Map

Sharing with community, the way they need it



Production PDF

Color Separation and Transparency



Color Mapping Rules (changed)

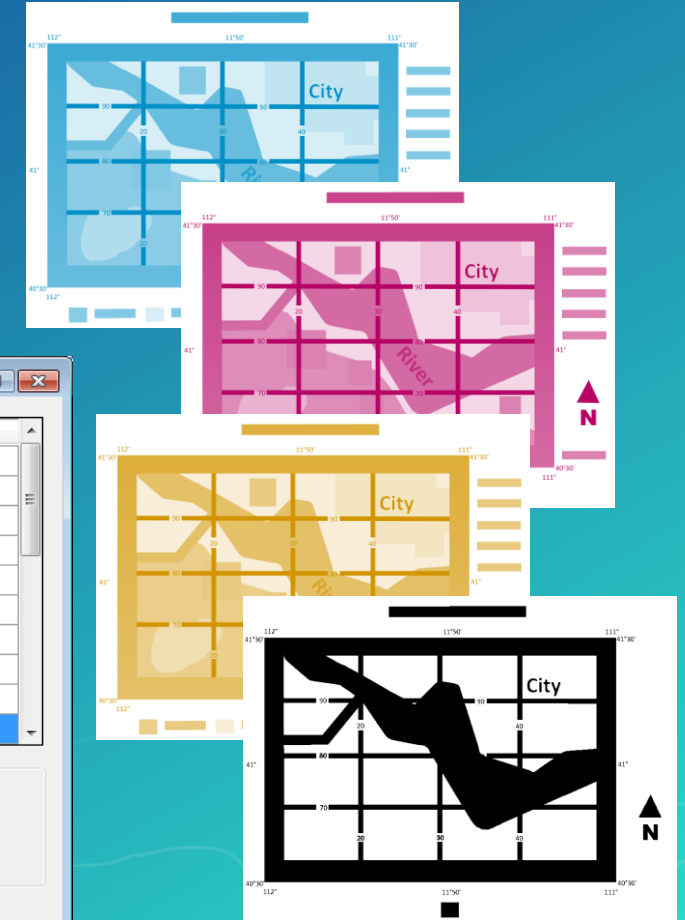
Input	Input Value	Output	Output Value	Tint %	Overprint
	RGB 255 255 255		CMYK 0 0 0 0	n/a	<input type="checkbox"/>
	RGB 0 0 0		CMYK 0 0 0 100	n/a	<input type="checkbox"/>
	RGB 153 0 0		CMYK 40 100 100 0	n/a	<input type="checkbox"/>
	RGB 255 0 0		CMYK 0 100 100 0	n/a	<input type="checkbox"/>
	RGB 230 0 0		CMYK 10 100 100 0	n/a	<input type="checkbox"/>
	RGB 221 232 198		CMYK 13 9 22 0	n/a	<input type="checkbox"/>
	RGB 118 145 59		CMYK 54 43 77 0	n/a	<input type="checkbox"/>
	RGB 255 247 191		CMYK 0 3 25 0	n/a	<input type="checkbox"/>
	RGB 230 200 0		Mustard 10 22 100 0	100	<input type="checkbox"/>
	RGB 205 233 239		CMYK 20 9 6 0	n/a	<input checked="" type="checkbox"/>

Current Map Review

Use this button to add colors from map to the table. Only colors that are not in the table will be shown.

Show preview for selected input color(s), vector graphics only. This starts the PDF viewing application registered in your system.

1 out of 24 selected Warning(s)



Automating Output

Using Python

```
if export == 'JPEG':
    filename = map_doc_name + ".jpg"
    outfile = os.path.join(outputdirectory, filename)

    # Run the export tool
    arcpy.mapping.ExportToJPEG(mxd, outfile)

elif export == "MAP PACKAGE":
    filename = map_doc_name + ".mpk"
    outfile = os.path.join(outputdirectory, filename)
    mxd = mxd.filePath

    # Run the export tool
    arcpy.PackageMap_management(mxd, outfile)

elif export == 'PRODUCTION PDF':
    filename = ap_doc_name + ".pdf"
    outfile = os.path.join(outputdirectory, filename)
    setting_file = os.path.join(product_location, "colormap.xml")

    arcpyproduction.mapping.ExportToProductionPDF(mxd, outfile, setting_file)

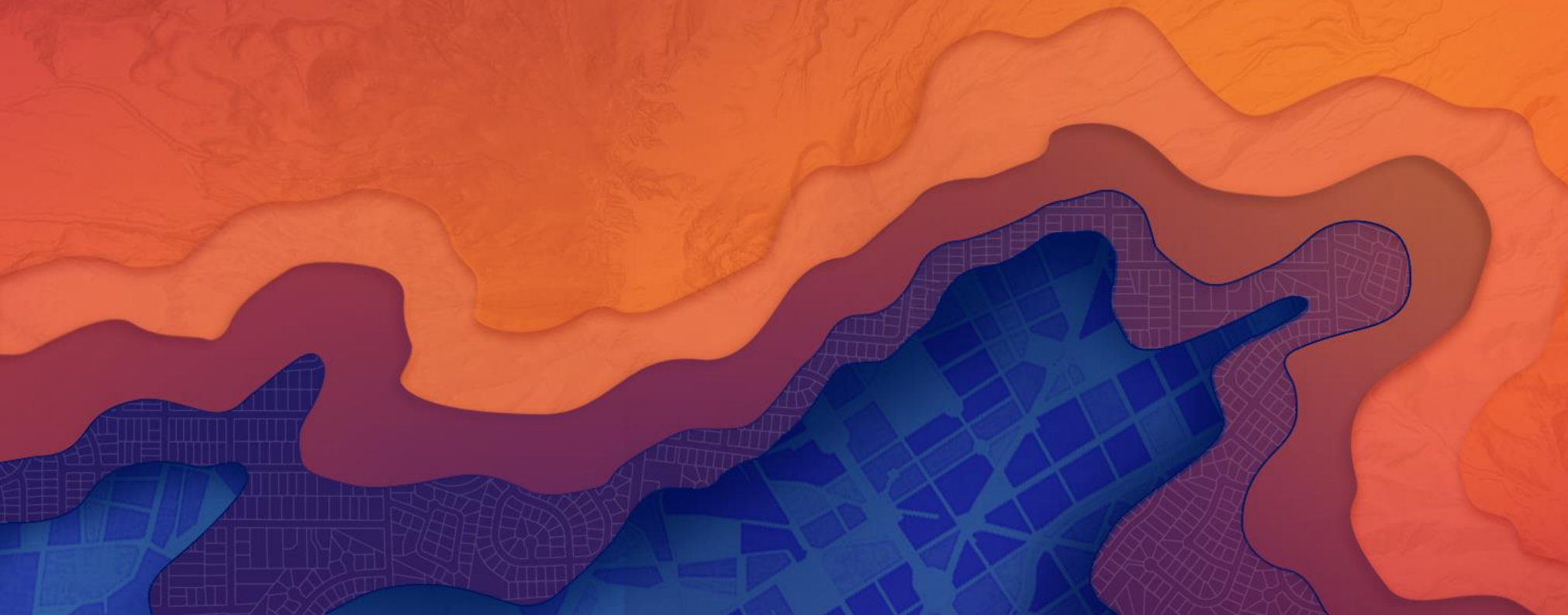
arcpy.AddMessage("Output is located: " + outfile)
```



Demo: Layout and Output

Joe Sheffield

Product On Demand

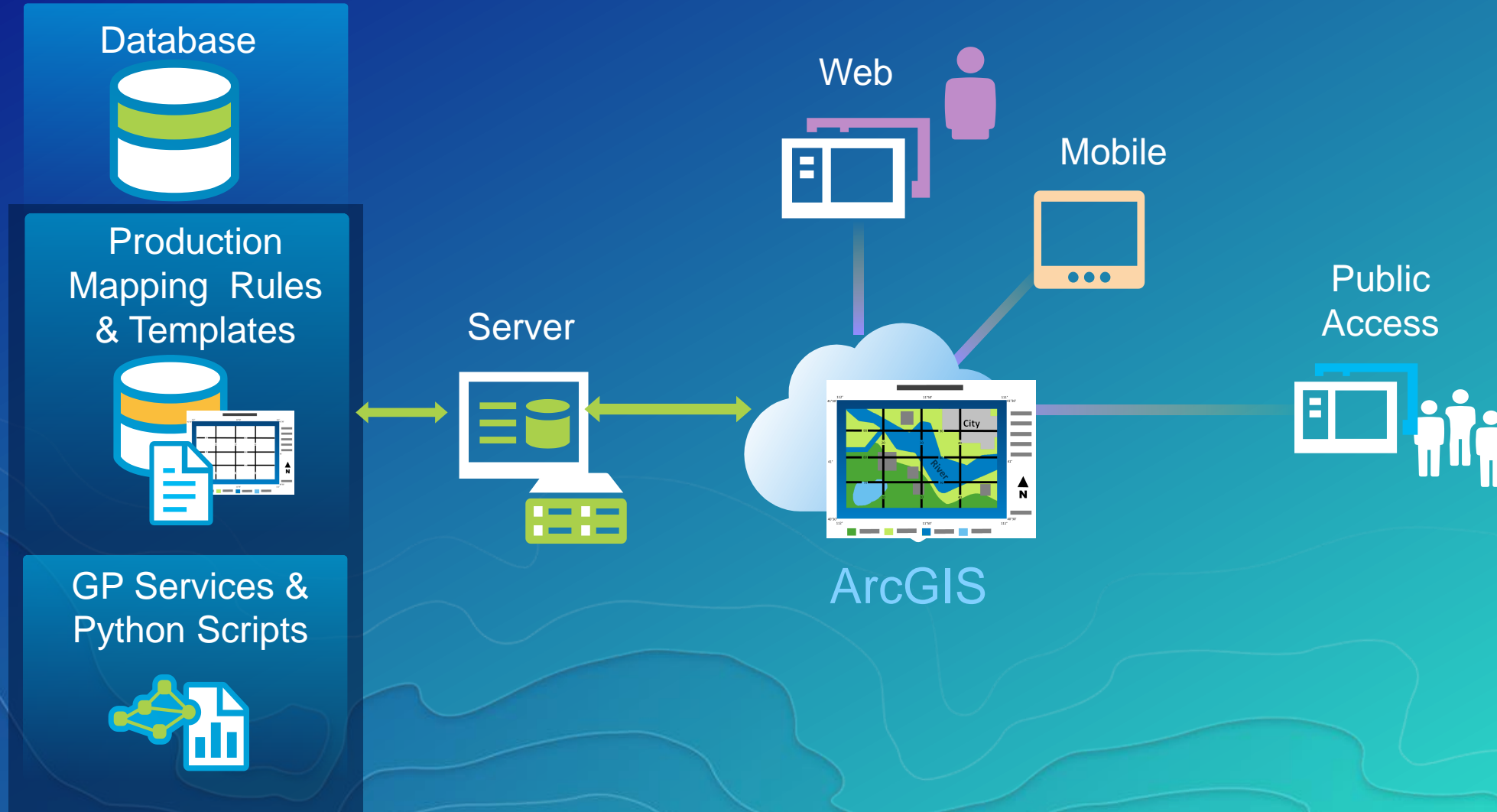


Production Mapping for ArcGIS Server

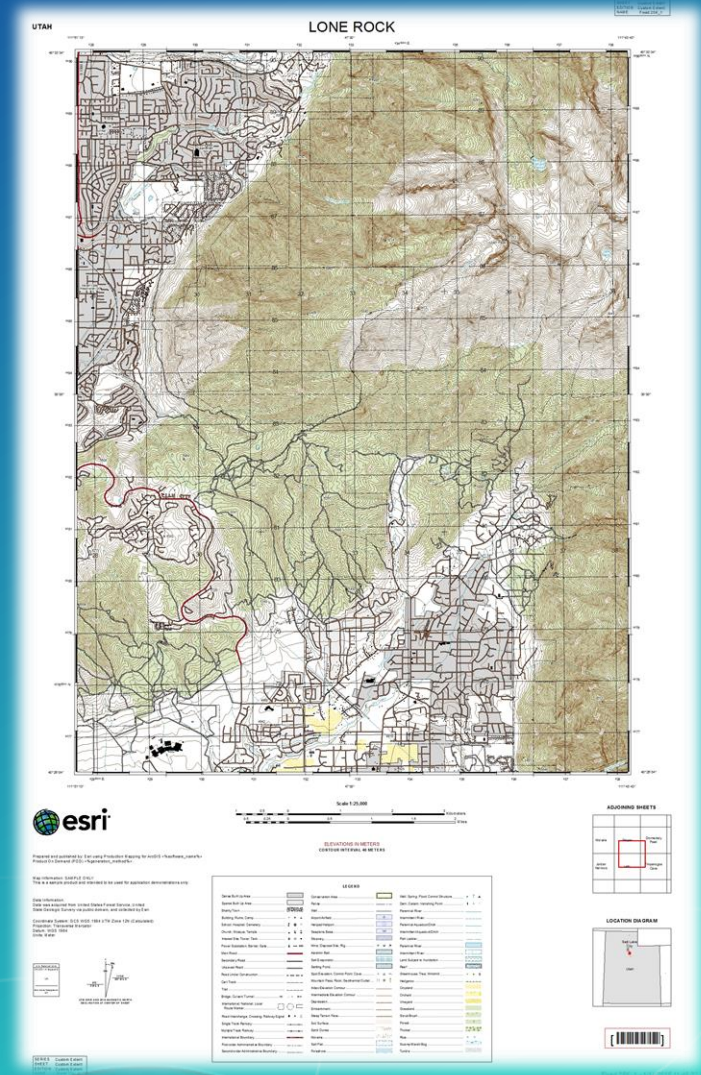
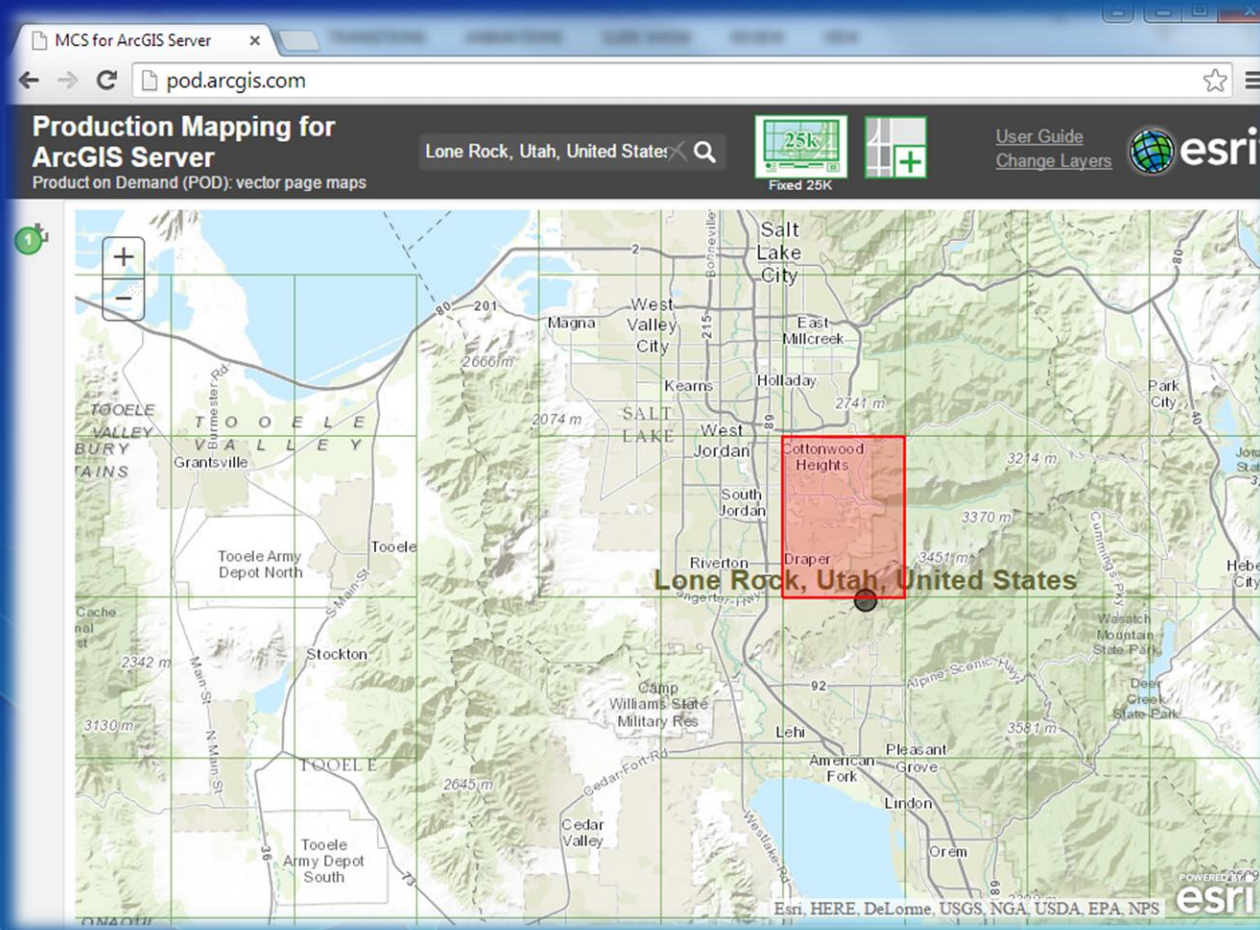
Enabling self-service mapping for Authoritative products



Production Mapping for ArcGIS Server



Product on Demand



<https://github.com/Esri/product-on-demand>



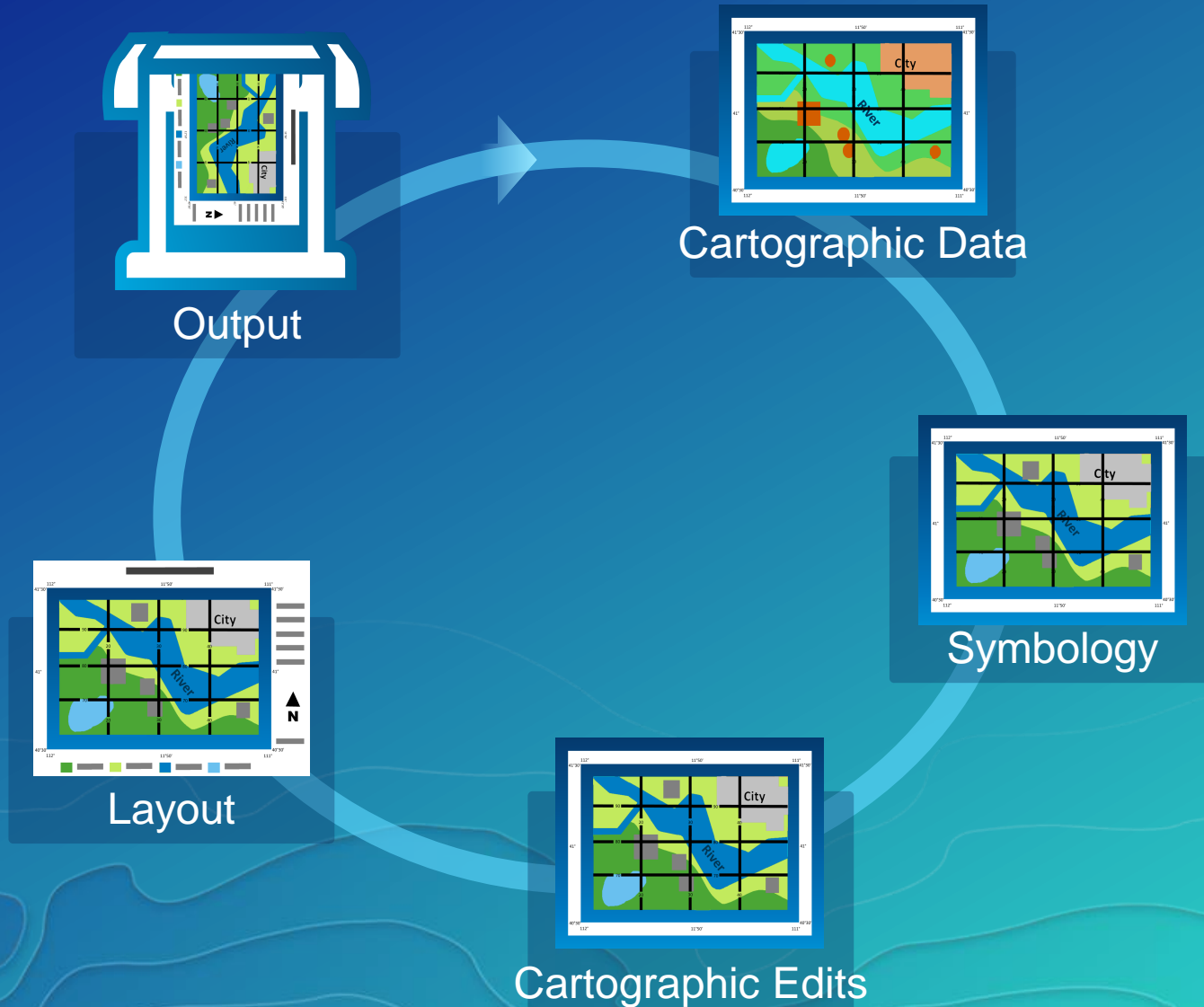
Demo: POD

Joe Sheffield

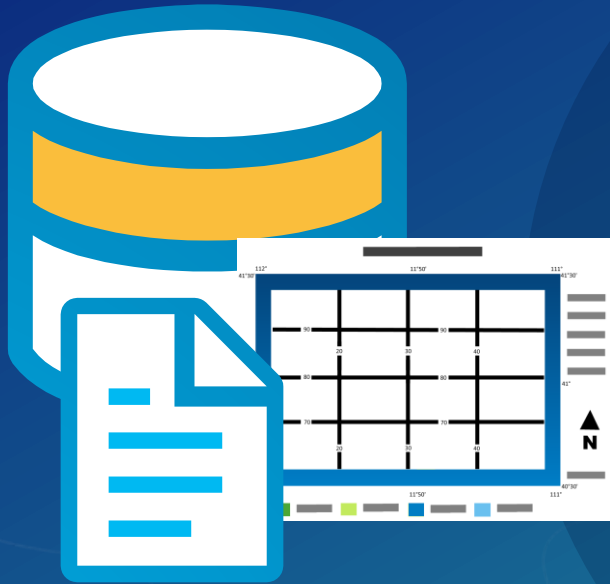
Conclusions



Map Automation & Advanced Cartography



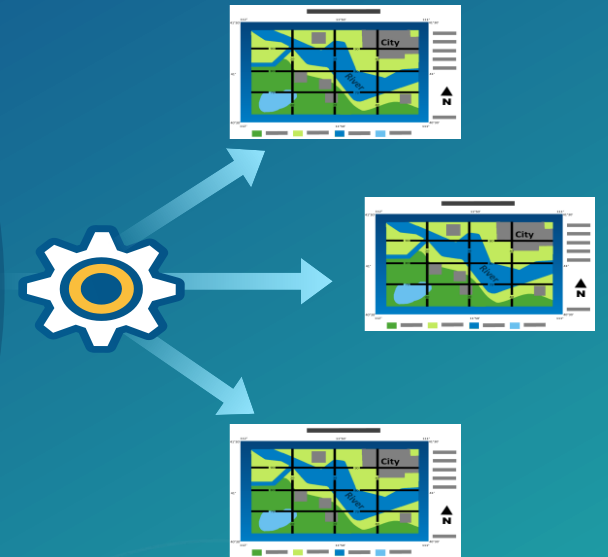
Map Automation



Production Mapping
Cartographic Rules



Geoprocessing
& Python



Standard
Output

References

Production Mapping

Learn More:

Desktop: <http://www.esri.com/productionmapping>

Server: <http://www.esri.com/software/arcgis/arcgisserver/extensions/production-mapping>

Email us: productionmapping@esri.com

CTM

Get it:

<https://github.com/esri/ctm>

Product on Demand

Get it:

<https://github.com/Esri/product-on-demand>

Try it:

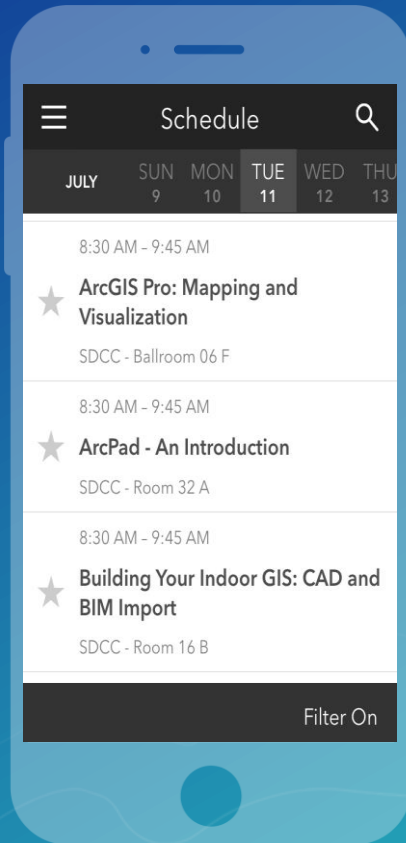
<http://pod.arcgis.com/>

Please Take Our Survey on the Esri Events App!

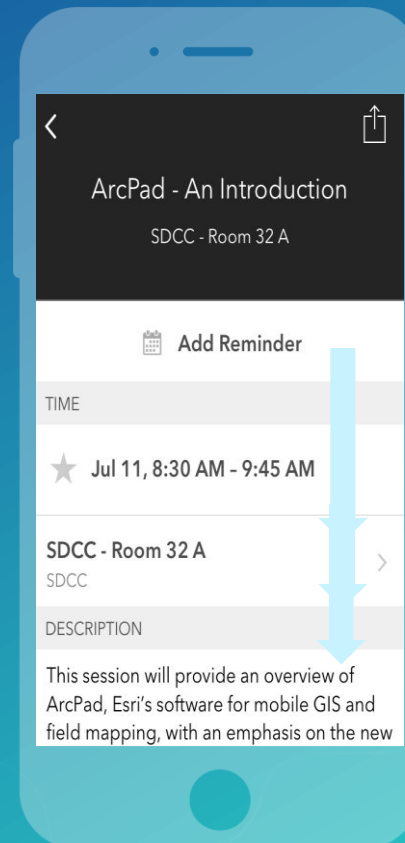
Download the Esri Events app and find your event



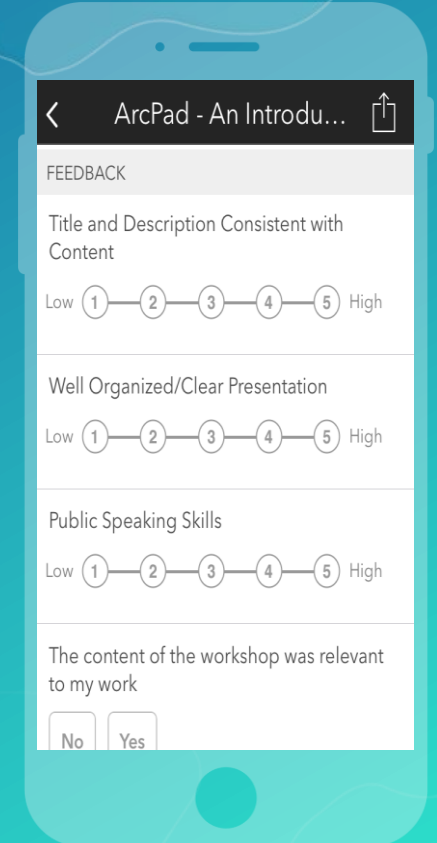
Select the session you attended



Scroll down to find the survey



Complete Answers and Select "Submit"





esri

THE
SCIENCE
OF
WHERE