Using ArcGIS to Develop a Fire Map Book

Or "How I Learned to

Love Symbol Levels"

Brian Bradshaw
GIS Project Manager
(858) 636-4880
bbradshaw@sddpc.org

Acknowledgements

- ## Map Book Editing Team
  - John Urata, Linzy Qi, Lynn Finkel

- ## One Way Road Symbology, Road Update Program
  - Ron Harris

- ## Customers & Supervisors
  - Susan Infantino, Laura Brenner, Jody Betry, Marlyn Webb

- ## Funding
  - Dept of Homeland Security Urban Area Strategic Initiative (UASI) Grant

Overview

- Description - Objectives & Challenges/Issues

- Advanced Road Symbology
  - Cased Roads, Symbol Levels, VBA & ArcObjects

- Other Advanced Symbology
  - One Way Road Arrows – Labels Using a Symbol Font
  - Gates - Symbol Rotations
  - Hydrant Offsets – VBA & ArcObjects

- Map Book Production
  - Map Book Definition & Page Layout
  - Helpful Map Document Settings
  - Production Workflow & Computing Environment

- VBA Code Samples

Description

- Replace a Hand Drawn Map Book Used by San Diego Fire-Rescue Dept. with a GIS-Based Map Book
  - Hardcopy Book on Vehicles to Backup GPS Navigation System
  - Large Wall Maps at Each Station for Area Overview

- Duplicate Same Features as Hand Drawn Map Book

- Replicate Hand Drawn Symbology

- Use Regional Symbology for Point & Poly Features

- Replicating Road Overpass/Underpass Symbology
  - Solved Using Cased Roads & Symbol Levels
  - Resulted in Unique Version of Road Data for Map Book

- Digital Data Issues
  - No Digital Data for Some Features or Feature Types
  - Some of Available Digital Data Not Complete or Accurate

- Designing Single Layout for Map Book & Wall Maps

- No Multi-User Editing Environment Available

Hand Drawn Map Book vs GIS-Based Map Book
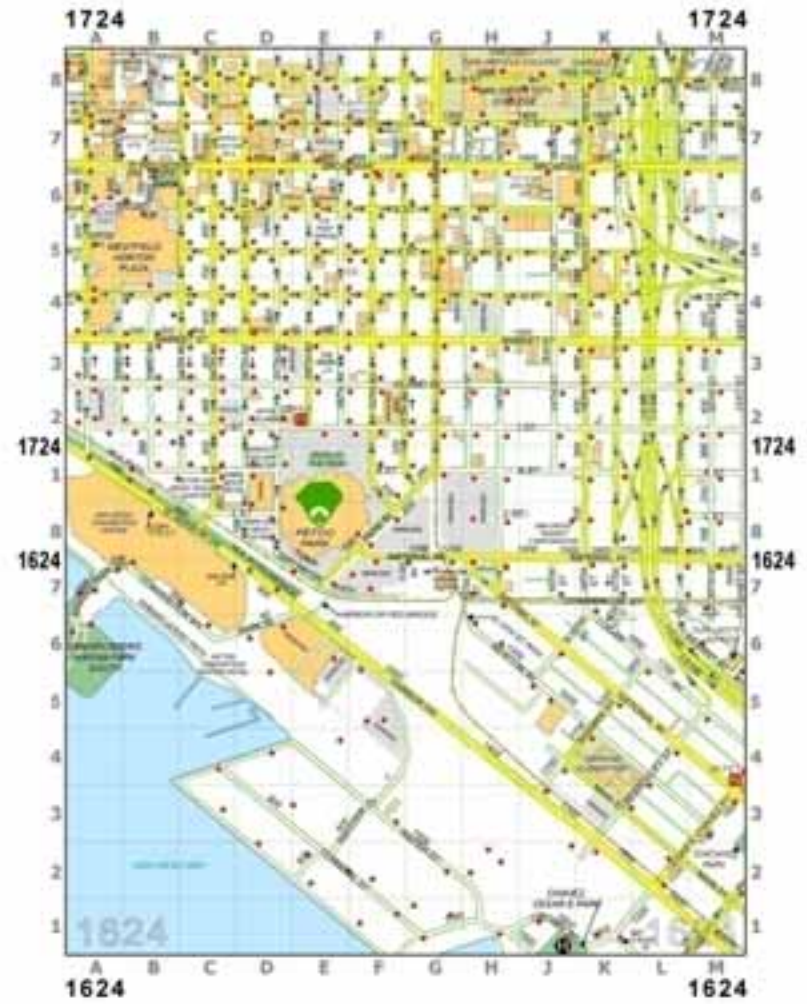
Hand Drawn Map Book vs GIS-Based Map Book
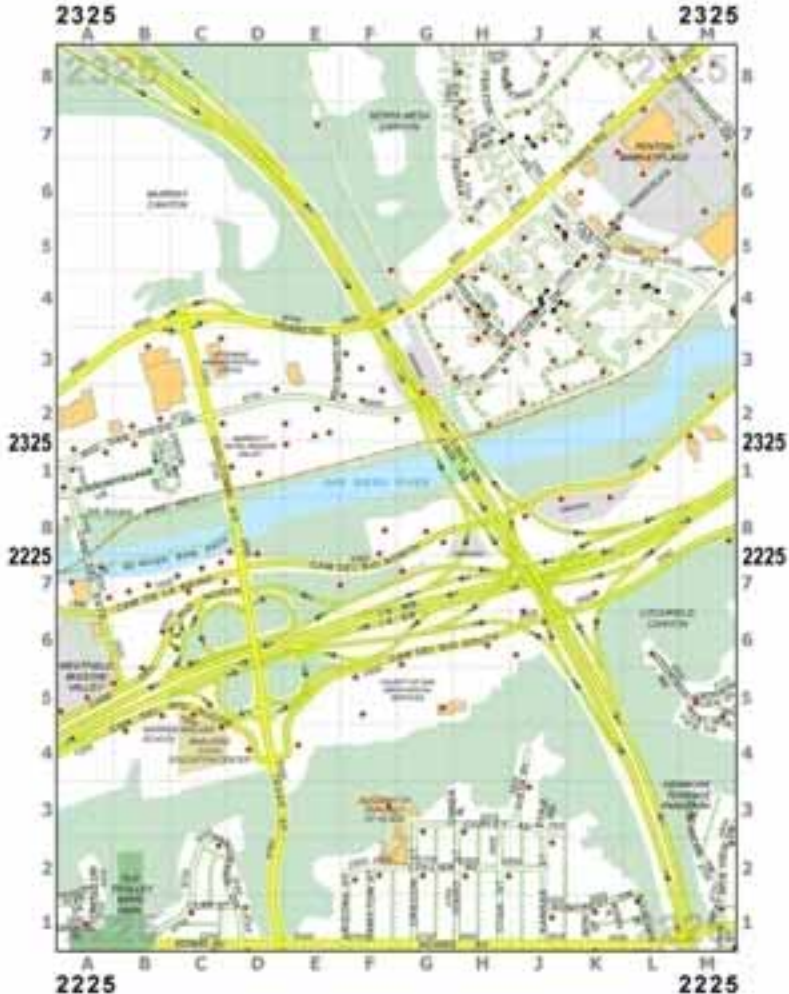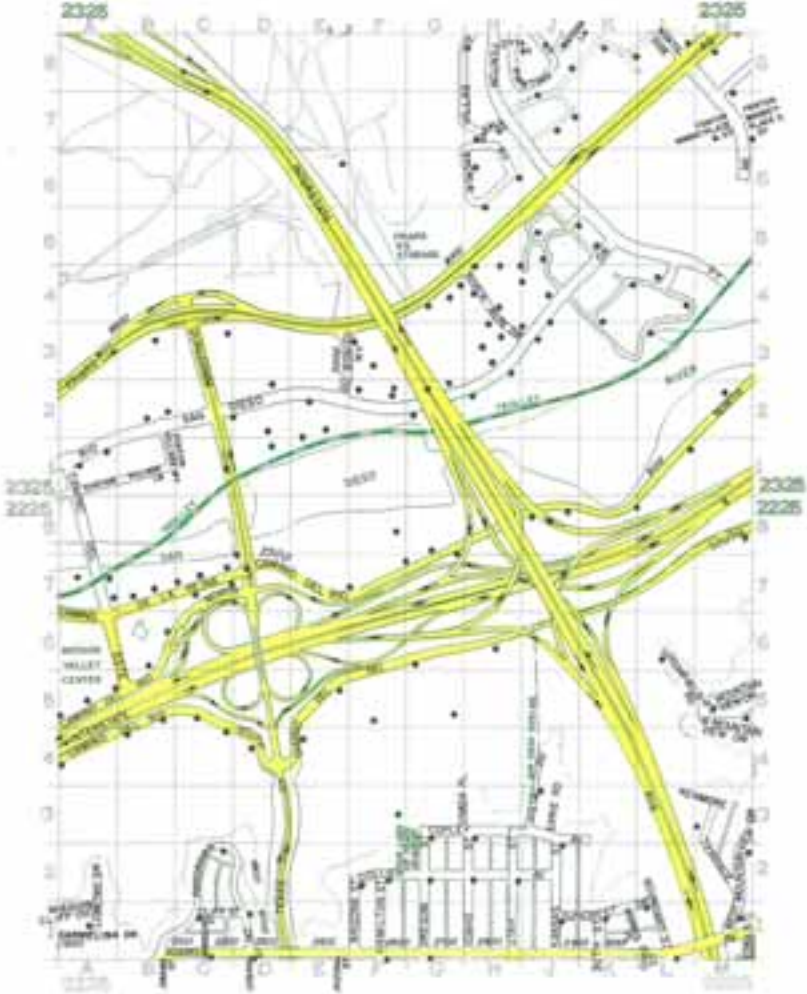
Advanced Road Symbology

- Basic Road Symbology
  - Cased Roads Using Unique Values with Many Fields

- Symbol Levels Description

- Symbol Levels Issues
  - Roads Data Format - From/To Node Elevations vs Segment Elevations

- Making Symbol Levels Work

- Automate Road Symbology Using VBA & ArcObjects

- Symbol Levels Alternative

- Cased Road Symbols
  - Cartographic Line Symbol with 2 Layers
    - Wider Bottom Layer Appears as Outline (Casing)
  - Specify Line Join = Round (Smooth Edges Between Segments)
  - Line Cap = Butt Creates Open End
    - Best to "Join" Road Segments to Show Connectivity
  - Line Cap = Round Creates Closed End
    - Best Representation for Cul De Sacs

- Unique Values with Many Fields (3 Fields)
  - Road Type (Freeway vs Major Rd vs Private Rd) Defines Width
  - Route Type (Route of Response or Not) Defines Symbol Color
  - Elevation Used to Define Symbol Level

Basic Road Symbology
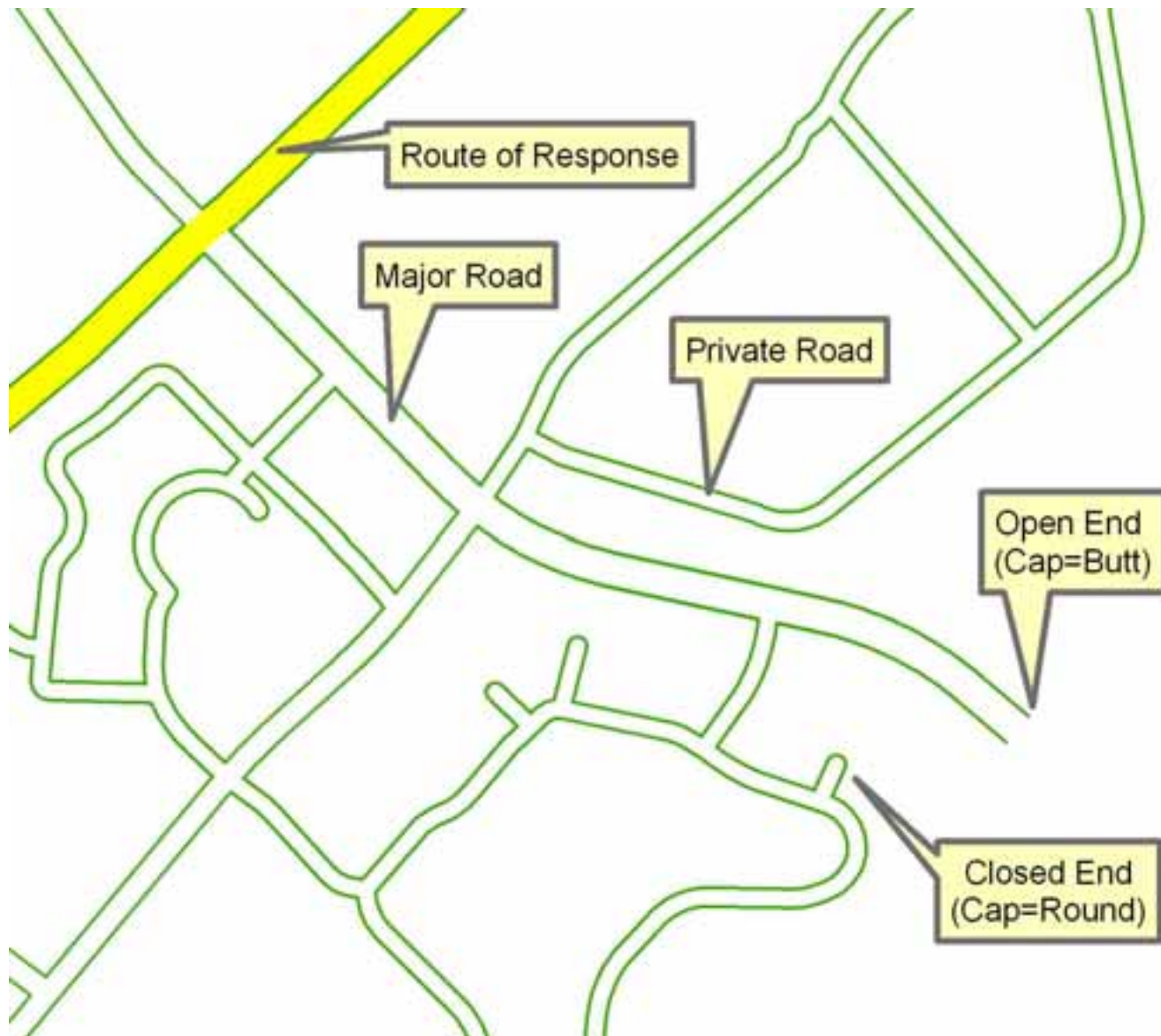
- Symbol Levels - Basic Concept
  - Overrides ArcMap Default Drawing Sequence
  - Can Specify Drawing Order of Individual Layers for Multilayer Symbols
  - Allows Symbology to "Blend" Where Streets Intersect or Not Blend for Overpass/Underpass
  - Can Use Elevations to Visually Separate Overlapping Road Segments
  - Roads with Same Elevation can be "Blended"

- # Symbol Levels Options
  - ## Default View
    - Specify Draw Order by Moving Symbols Up/Down List
    - Specify Join to Blend All Features With That Symbol
    - Specify Merge to Blend Symbol With Symbol Above

  - ## Advanced View
    - Specify Numeric Values for Each Symbol Layer to Define Blending (Each Symbol Layer Has a Value)
    - Symbol Layers with Same Number (Level) Will Blend
    - Higher Numbers (Levels) Drawn Over Lower Numbers
    - Specify Same Number for Outer Symbol Layer for Roads at Same Elevation
    - Specify Higher Numbers for Roads with Higher Elevations

Symbol Levels – Advanced View

- Limited Documentation and Samples

- Data Format - Node Elevations vs Segment Elevation
  – Elevation Defined for From Node & To Node, NOT Segment
  – Symbol Levels Works Best With a Single Elevation per Segment
  – Majority From & To Nodes Have Same Elevation
    - 155875 Road Segments Total
    - 1891 Segments Where From Node Elevation NE To Node Elevation

- Performance Issues
  – Partial Drawing & Then Hangs Up Using Shapefile
  – Slow But Acceptable Performance with File Geodatabase

- Use Open Ends Where Possible
  - Keep Closed End for Local Streets & Roads with Elevation = 1
  - ***Eliminates Lines Between Segments Joined End-to-End, Regardless of Elevation Differences
  - Still Have Lines Between Segments Joined at an Angle

- Define & Adjust Segment Elevations
  - Begin with Segment Elevation = Max of From or To Elevation
  - Change Segments Joined at Angle to Same Elevation
  - Eliminates Most Remaining Lines Between Segments

- Split Roads For "3 Segment Triangle Problem"
  - ***3 Roads Forming Triangle at Overpass Requires 4 Segments

Road Symbology – Initial Results
(Closed Ends, Elevation=Max of From & To Els)

Road Symbology – Road Ends Adjusted
(Open Ends for Elevations NE 1, Not Local Street)
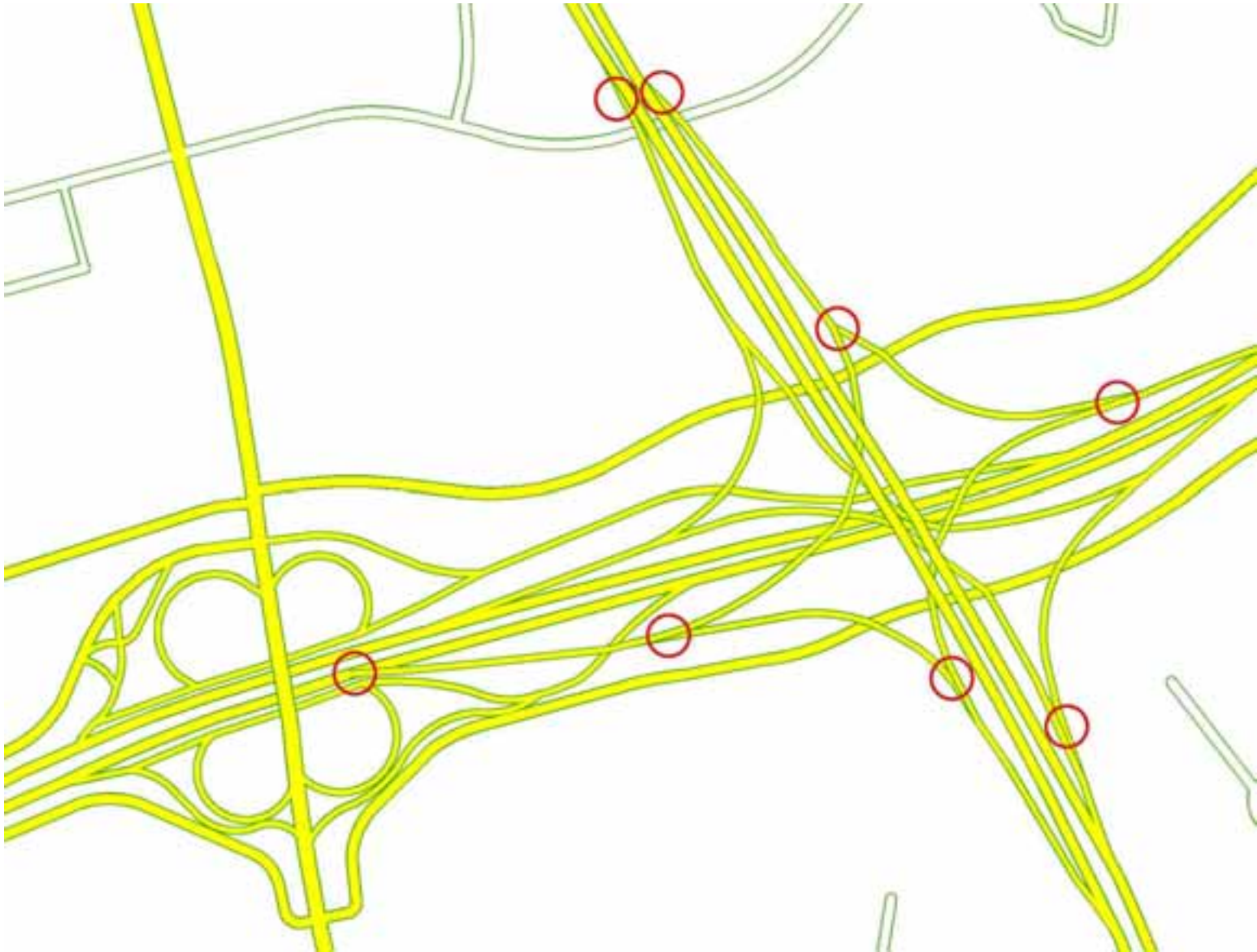
Road Symbology
Elevation Adjustment – After
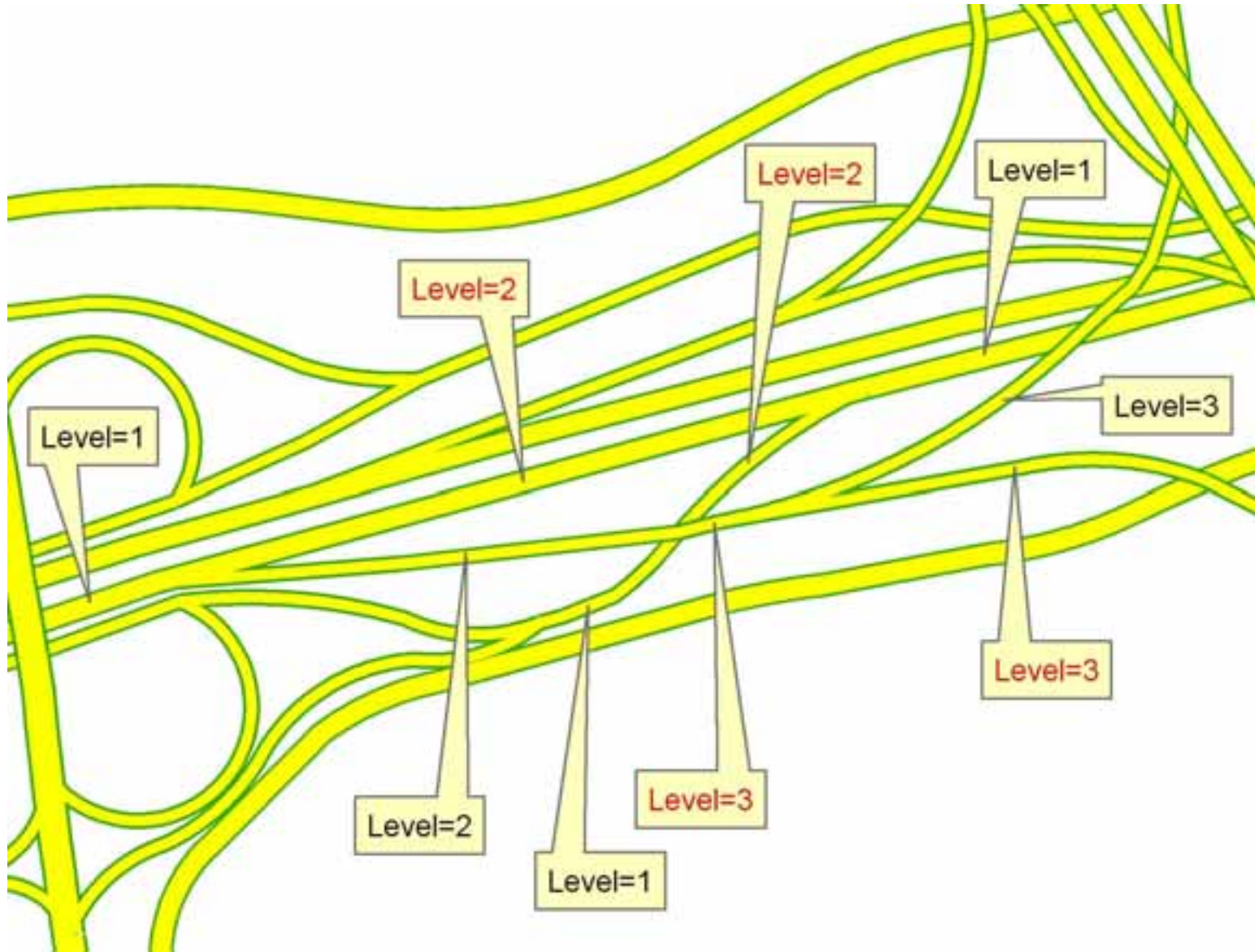
Road Symbology – Elevations Adjusted
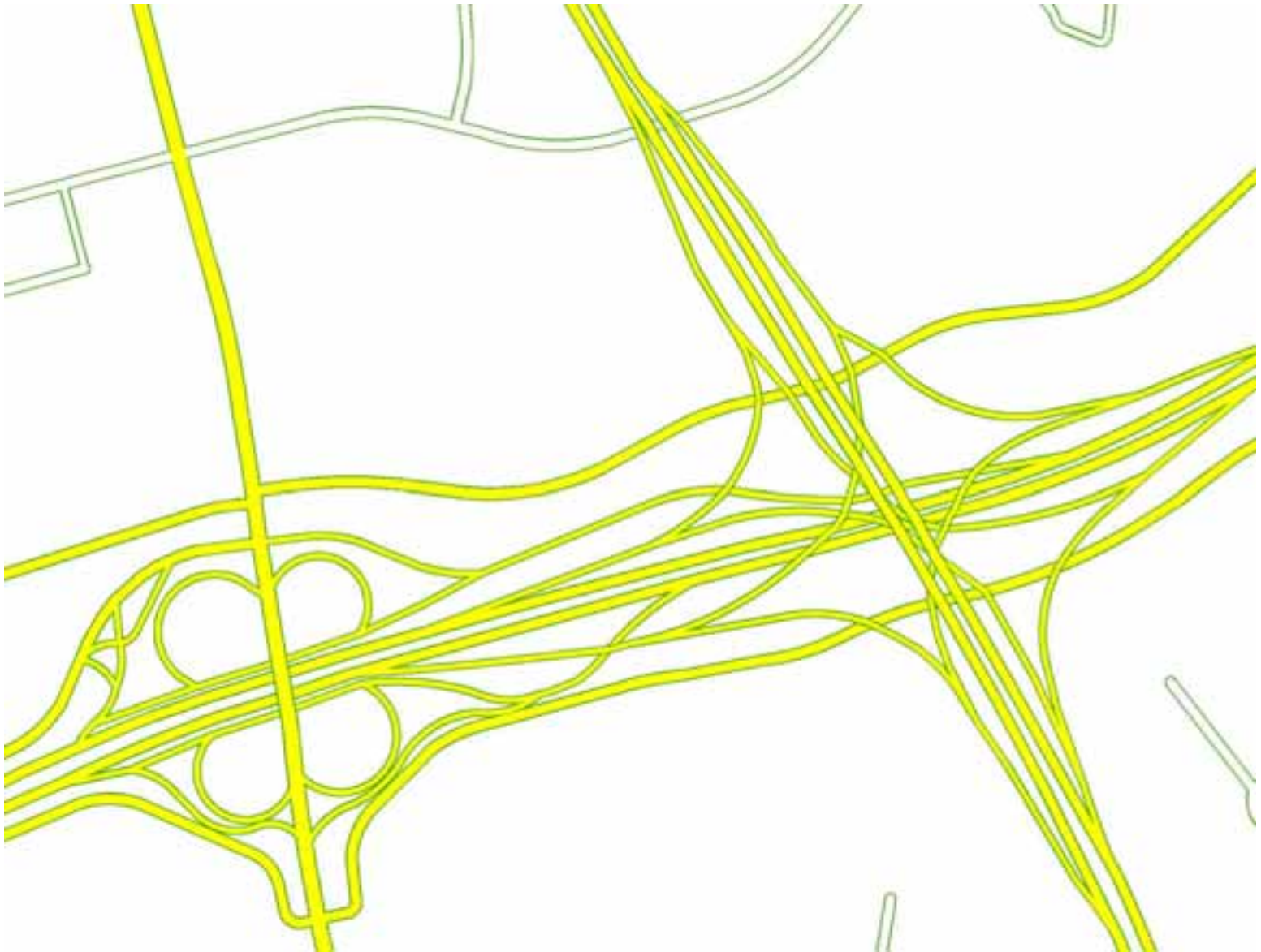
Road Symbology – Elevations Adjusted
3 Segment Triangle Problem Remains

3 Segment Triangle – Before Road Split

Level=2

Level=1

Level=1
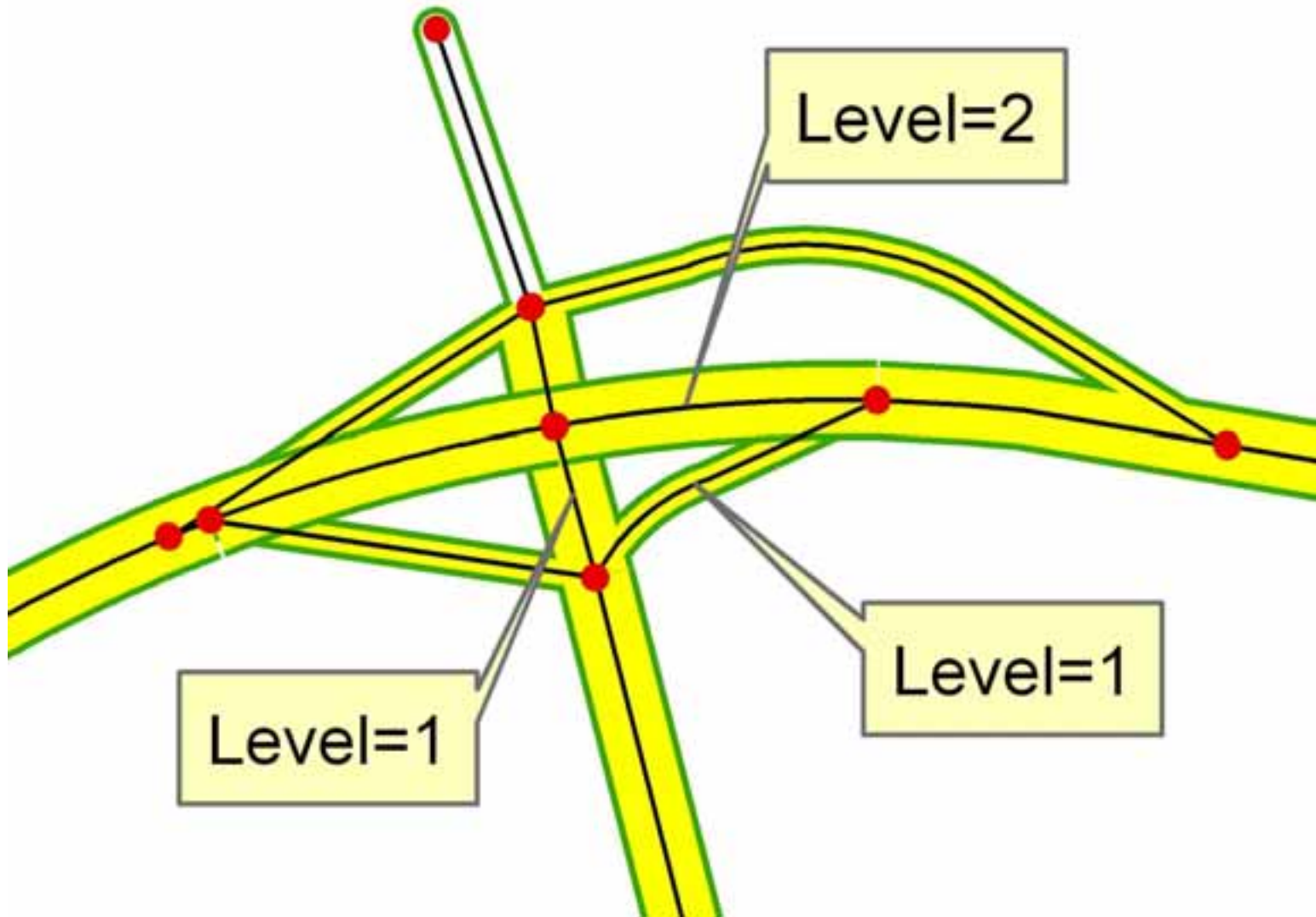
3 Segment Triangle – After Road Split (Creating 4 Segment Triangle)
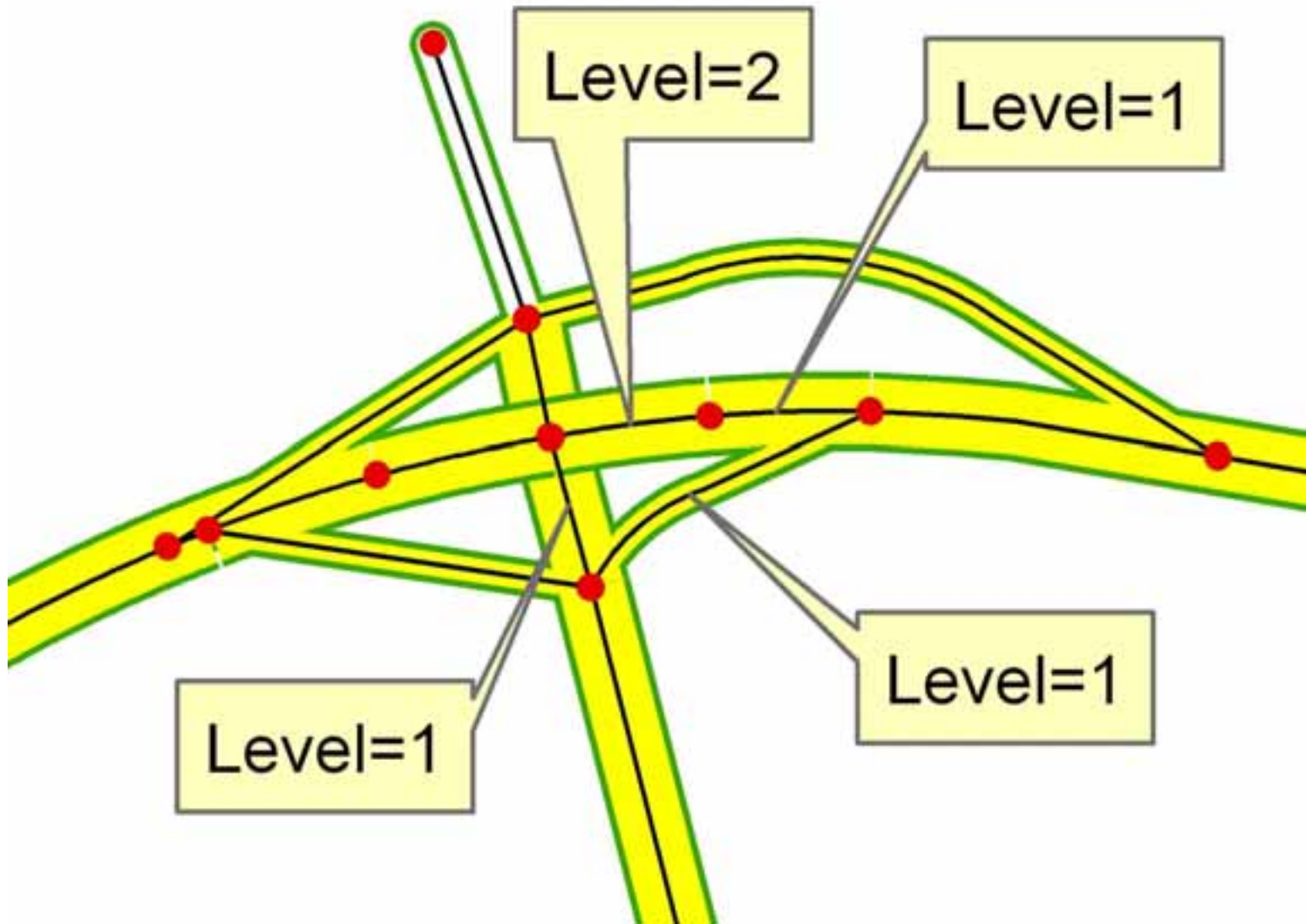
Road Symbology – Final Results

Automate Road Symbology Using VBA & ArcObjects

- 73 Different Road Symbols (Combos of Width, Color, Elevation)

- Issue - Limited Documentation and Sample Code

- Code Summary
  - Define Map Layer to Symbolize
  - ISymbolLevel: Specify to Use Symbol Levels for Map Layer
  - ICartographicLineSymbol: Define Symbol Layers (Color, Width, Cap)
  - IMultiLayerLineSymbol: Group Symbol Layers
  - IUniqueValueRenderer: Assign Attribute Values to Specific Symbol
  - IMapLevel: Define Symbol Levels to Symbol Layers
  - *** Specify MapLevel = -1 for MultiLayer Symbol, Then Specify Actual Symbol Levels for 2 Separate Layers of Cased Road

- ESRI Tech Support Solution
  - "Can't use Symbol Levels"
  - Use Cartographic Representations
    - Lengthy Multi Step Process
    - Requires Great Deal of Manual Interaction
    - Creates Separate Graphic Over Roads to Simulate Overpass
      - Graphic Does Not Change if Road Data Modified
    - Slows Draw Speed

Other Advanced Symbology

- Symbolized as Labels Using Symbol Font
  - Wingdings 3 vs Alphabet Font (Arial, Times, etc.)

- Code Road Segment as Letter Position that Matches One Way Arrow Symbol/Direction
  - ← is 6th Symbol (code as f)
  - → is 7th Symbol (code as g)
  - ←→ is 14th Symbol (code as n)

- Requires Maplex
  - *** "Align Label to Direction of Line"
    - Placement Properties → Label Position → Orientation → Label Alignment

Wingdings 3 (OpenType)

OpenType Font, Digitally Signed, TrueType Outlines
Typeface name: Wingdings 3
File size: 35 KB
Version: Version 1.55
Wingdings 3 designed by Bigelow & Holmes Inc. for Microsoft Corporation. Copyright ©
Reserved.

⇨⇦⇨⇦⇨←→↑↓↖↗↙↘↔↕▲▼△▽◀▶◁▷◣◢ ↳↰↱↴↵↥↧⇐⇒↟↡⌒⌒↺↻⌐⌐⇧⇧⇦ ↔↕⋯⟶↥↧↯↵↲↳↘↰↱↵↴↲⇁↼↔⇃↗↤
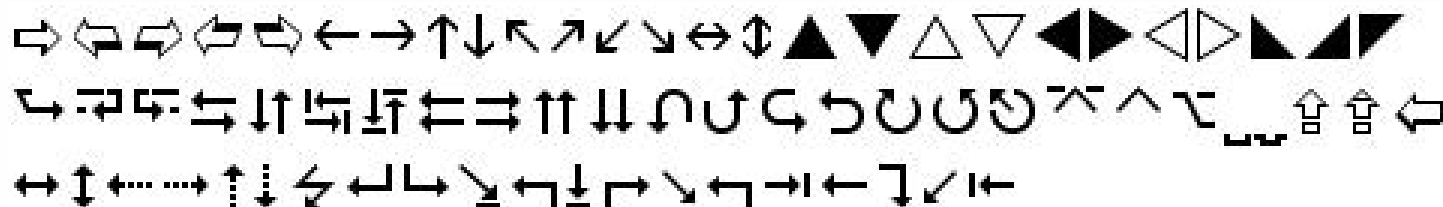
- Advanced Symbol Option in ArcMap

- Specify Field to Define Angle

- Specify Angle for Each Feature in Attribute Table

- Visually Verify Angle During Editing

Hydrant Offsets

- Description
  - Road Symbology Makes Some Hydrants Appear In Street
  - Older Version of Hydrant Data Manually Offset To Address This
  - Take Advantage of Offsets from Older Data While Using New Data Set

- Calculate Offset Values for Each Individual Hydrant
  - Calculate X, Y Position of Each Feature in Old & New Datasets
    - Use Calculate Geometry Function from Attribute Table Options
  - Join Tables & Calculate Difference in X,Y Position to Identify X,Y Offset

- ArcMap Symbol Settings Only Enables Single Offset Value for Entire Feature Class/Map Layer vs Each Feature Like Symbol Rotation

- Use VBA Code to Apply Calculated Offsets to Each Feature

Map Book Production

- ESRI Developer Sample – DSMapBook
  - Specify Grid File to Define Individual Pages
  - Adds Map Book Tab to Table of Contents with List of Pages in Grid
    - Provides Quick Access to Each Page
  - Allows Layout Customization Used for All Pages
    - Page Specific Items (Page #) Use Attribute in Grid File
  - Built In Functionality with ArcGIS 10

- Layout Issues
  - Map Page & Subgrid Labels on Outside of Map Data for Map Book
  - Need Map Page Label on Map Data for Wall Maps
    - Used Large Outline Font with Light Gray Color to Simulate Transparency
  - Single Data Extent for 8.5x11 Book Page and 15x20 Wall Maps

- Definition Queries
  - Subset Data Without Editing Source Data
  - Easily Modified During Map Production As Needed

- Label Expressions
  - Eliminate Undesired Labels without Editing Source Data
  - Replace Incorrect Labels/Typos without Editing Source Data

- Maplex Label Settings
  - Superior Control over Label Placement
  - Helps Minimize Annotation Adjustment

- Feature Linked Annotation
  - Helpful During Annotation Editing to "Follow Feature"

- Production Concept
  - Finalize Data Creation & Editing Before Individual Page Production
  - Master File Geodatabase & Map Document with Duplicate GDB's & MXD's for Each Editor, 4 Editors
  - Editors Create Feature Linked Annotation One Page At a Time
    - Converted All Labels to Annotation
  - Each Editors GDB Same Except Annotation Created for Their Pages

- Production Issues
  - Errors in Road & Other Data Required Editing During Page Production
    - Significantly Impacted Schedule
    - Required Iterative Editing of Master GDB & Manual Updates to Editor GDBs
  - Changes in Symbology, Label Settings Required Editing MXDs During Production

- Editing on Desktop Computers

  - ArcMap 9.3 SP1 & Maplex Extension

  - Windows XP SP3

  - 1.6GHz Intel Pentium Dual Processor

  - 2 GB RAM

- Data Stored on Network Server

  - 1 GB Ethernet Connection

  - Master & Editor Geodatabases

VBA Code Samples

Road Symbology
Sample VBA Code

```
Public Sub Symbolize_Roads()

' define layer
Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap
Dim pLayer As ILayer
Set pLayer = pMap.Layer(0)
Dim pFLayer As IFeatureLayer
Set pFLayer = pLayer
Dim pGFLayer As IGeoFeatureLayer
Set pGFLayer = pFLayer
Dim pFClass As IFeatureClass
Set pFClass = pGFLayer.FeatureClass

'define unique value renderer
Dim pUVRen As IUniqueValueRenderer
Set pUVRen = New UniqueValueRenderer
Set pGFLayer.Renderer = pUVRen
'specify fields to render unique values with
pUVRen.FieldCount = 3
pUVRen.Field(0) = "SEGCLASS"   'Road type attribute
pUVRen.Field(1) = "RteType"    'Route of Response attribute
pUVRen.Field(2) = "FMB_ELEV"   'Elevation attribute
pUVRen.FieldDelimiter = ","
pUVRen.UseDefaultSymbol = False
```

```vba
' set up symbol levels
Dim pSymLevels As ISymbolLevels
Set pSymLevels = pGFLayer
pSymLevels.UseSymbolLevels = True

'define colors
Dim pYlwColor As IRgbColor
Set pYlwColor = New RgbColor
pYlwColor.RGB = RGB(255, 255, 0)
Dim pGrnColor As IRgbColor
Set pGrnColor = New RgbColor
pGrnColor.RGB = RGB(56, 168, 0)
Dim pWhtColor As IRgbColor
Set pWhtColor = New RgbColor
pWhtColor.RGB = RGB(255, 255, 255)

'define widths
Dim d32Width As Double
d32Width = 3.2
Dim d42Width As Double
d42Width = 4.2
```

```vba
'Define symbols for Symbol 400
Dim pWht400CLSym As ICartographicLineSymbol
Set pWht400CLSym = New CartographicLineSymbol
Dim pGrn400CLSym As ICartographicLineSymbol
Set pGrn400CLSym = New CartographicLineSymbol
'Define layer 1
pWht400CLSym.Color = pWhtColor
pWht400CLSym.Width = d32Width
pWht400CLSym.Cap = esriLCSButt
pWht400CLSym.Join = esriLJSRound
'Define Layer 2
pGrn400CLSym.Color = pGrnColor
pGrn400CLSym.Width = d42Width
pGrn400CLSym.Cap = esriLCSButt
pGrn400CLSym.Join = esriLJSRound
'Add layers to MultiLayer Symbol - level 0
Dim p400MLSym As IMultiLayerLineSymbol
Set p400MLSym = New MultiLayerLineSymbol
p400MLSym.AddLayer pGrn400CLSym
p400MLSym.AddLayer pWht400CLSym
'add the unique symbols to the renderer - level 0
pUVRen.AddValue "4,2,0", "", p400MLSym
pUVRen.AddValue "5,2,0", "", p400MLSym
' define maplevels - level 0
Dim pMapLevel As IMapLevel
Set pMapLevel = p400MLSym
pMapLevel.MapLevel = -1
Set pMapLevel = p400MLSym.Layer(0)
pMapLevel.MapLevel = 2
Set pMapLevel = p400MLSym.Layer(1)
pMapLevel.MapLevel = 1
```

```vba
'Define symbols for Symbol 401
Dim pWht401CLSym As ICartographicLineSymbol
Set pWht401CLSym = New CartographicLineSymbol
Dim pGrn401CLSym As ICartographicLineSymbol
Set pGrn401CLSym = New CartographicLineSymbol
'Define layer 1
pWht401CLSym.Color = pWhtColor
pWht401CLSym.Width = d32Width
pWht401CLSym.Cap = esriLCSRound
pWht401CLSym.Join = esriLJSRound
'Define Layer 2
pGrn401CLSym.Color = pGrnColor
pGrn401CLSym.Width = d42Width
pGrn401CLSym.Cap = esriLCSRound
pGrn401CLSym.Join = esriLJSRound
'Add layers to MultiLayer Symbol - level 1
Dim p401MLSym As IMultiLayerLineSymbol
Set p401MLSym = New MultiLayerLineSymbol
p401MLSym.AddLayer pGrn401CLSym
p401MLSym.AddLayer pWht401CLSym
'add the unique symbols to the renderer - level 1
pUVRen.AddValue "4,2,1", "", p401MLSym
pUVRen.AddValue "5,2,1", "", p401MLSym
' define maplevels - level 1
Set pMapLevel = p401MLSym
pMapLevel.MapLevel = -1
Set pMapLevel = p401MLSym.Layer(0)
pMapLevel.MapLevel = 24
Set pMapLevel = p401MLSym.Layer(1)
pMapLevel.MapLevel = 14

' refresh map
pMxDoc.ActiveView.Refresh
pMxDoc.UpdateContents

End Sub
```

```vba
Public Sub SymoblizeHydrants()

Dim pMxDoc As IMxDocument
Set pMxDoc = ThisDocument
Dim pMap As IMap
Set pMap = pMxDoc.FocusMap
' define layer
Dim sLayer As String
sLayer = "wtr_hydrant_FMB"
Dim pLayer As IDataLayer
Set pLayer = FindLayerInTOC(sLayer)
Dim pFLayer As IFeatureLayer
Set pFLayer = pLayer
Dim pGFLayer As IGeoFeatureLayer
Set pGFLayer = pFLayer
Dim pFClass As IFeatureClass
Set pFClass = pGFLayer.DisplayFeatureClass

'define unique value renderer
Dim pUVRen As IUniqueValueRenderer
Set pUVRen = New UniqueValueRenderer
'specify field to render unique values with
pUVRen.FieldCount = 2
pUVRen.Field(0) = "OFFSET_X"  'for use without joins
pUVRen.Field(1) = "OFFSET_Y"  'for use without joins
pUVRen.FieldDelimiter = ","
pUVRen.UseDefaultSymbol = False
```

```vba
'Create the Simple Marker
Dim pCharMarker As ICharacterMarkerSymbol
'define the character font set
Dim pFont As IFontDisp
Set pFont = New StdFont
pFont.Name = "ESRI Default Marker"
'Create a red RGB color for the symbol
Dim pRGBClr As IRgbColor
Set pRGBClr = New RgbColor
pRGBClr.Red = 255

'retrieve x & y offset values and cycle thru each feature
Dim pFCursor As IFeatureCursor
Dim pFeat As IFeature
Set pFCursor = pFClass.Search(Nothing, False)
Set pFeat = pFCursor.NextFeature
Do Until pFeat Is Nothing
   ' define the marker symbol
   Set pCharMarker = New CharacterMarkerSymbol
   pCharMarker.size = 7
   pCharMarker.Color = pRGBClr
   pCharMarker.Font = pFont
   pCharMarker.CharacterIndex = 33
   pCharMarker.XOffset = pFeat.Value (pFClass.FindField ("OFFSET_X")) / 12
   pCharMarker.YOffset = pFeat.Value (pFClass.FindField ("OFFSET_Y")) / 12
   'add the unique symbol to the renderer
   pUVRen.AddValue pFeat.Value (pFClass.FindField ("OFFSET_X")) & "," & pFeat.Value (pFClass.FindField ("OFFSET_Y")), "", pCharMarker  'for use without joins
   Set pFeat = pFCursor.NextFeature
Loop
' assign the renderer to the layer
Set pGFLayer.Renderer = pUVRen

' refresh map
pMxDoc.UpdateContents
pMxDoc.ActiveView.Refresh

End Sub
```