



# Developing with the ArcGIS Cartographic APIs

*Craig Williams, Michael Grossman,  
Jeremy Wright*

# Overview

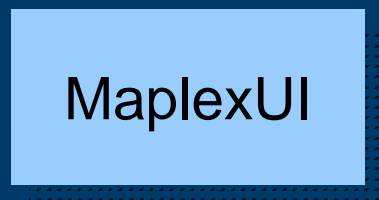
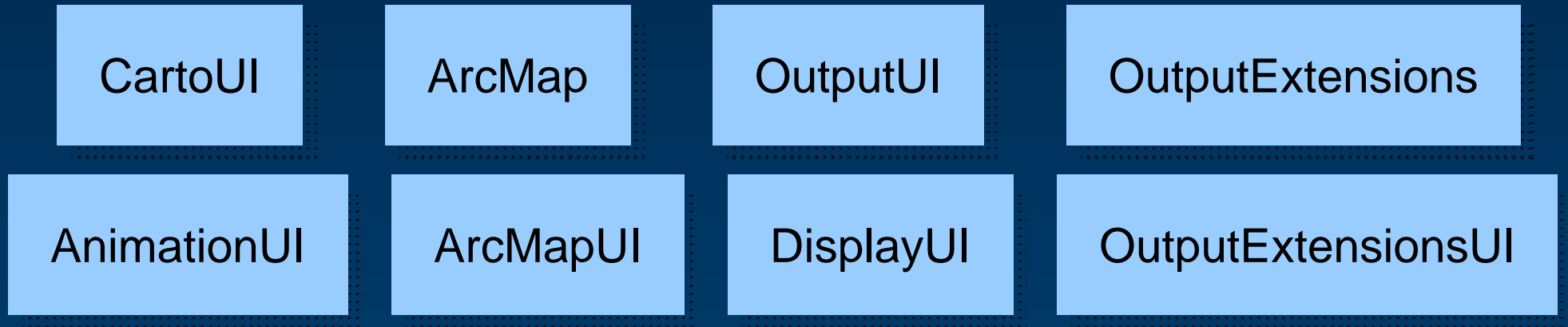
- **Libraries and Availability**
- **Cartography objects for**
  - Documents and Views
  - Maps and Layouts
  - Layers and Renderers
  - Cartographic Representations and Effects
  - Labeling
  - Elements
  - Display
  - Animation
  - Dynamic Display
  - Printing and Exporting
- **Demo 1 – Sample Engine Application**
- **Cartography objects for**
  - Symbols
  - Styles
- **Recommended Customizations**
- **Demo 2 – Sample Custom Geometric Effect**
- **Question period**

# Cartography API Assemblies

## Engine

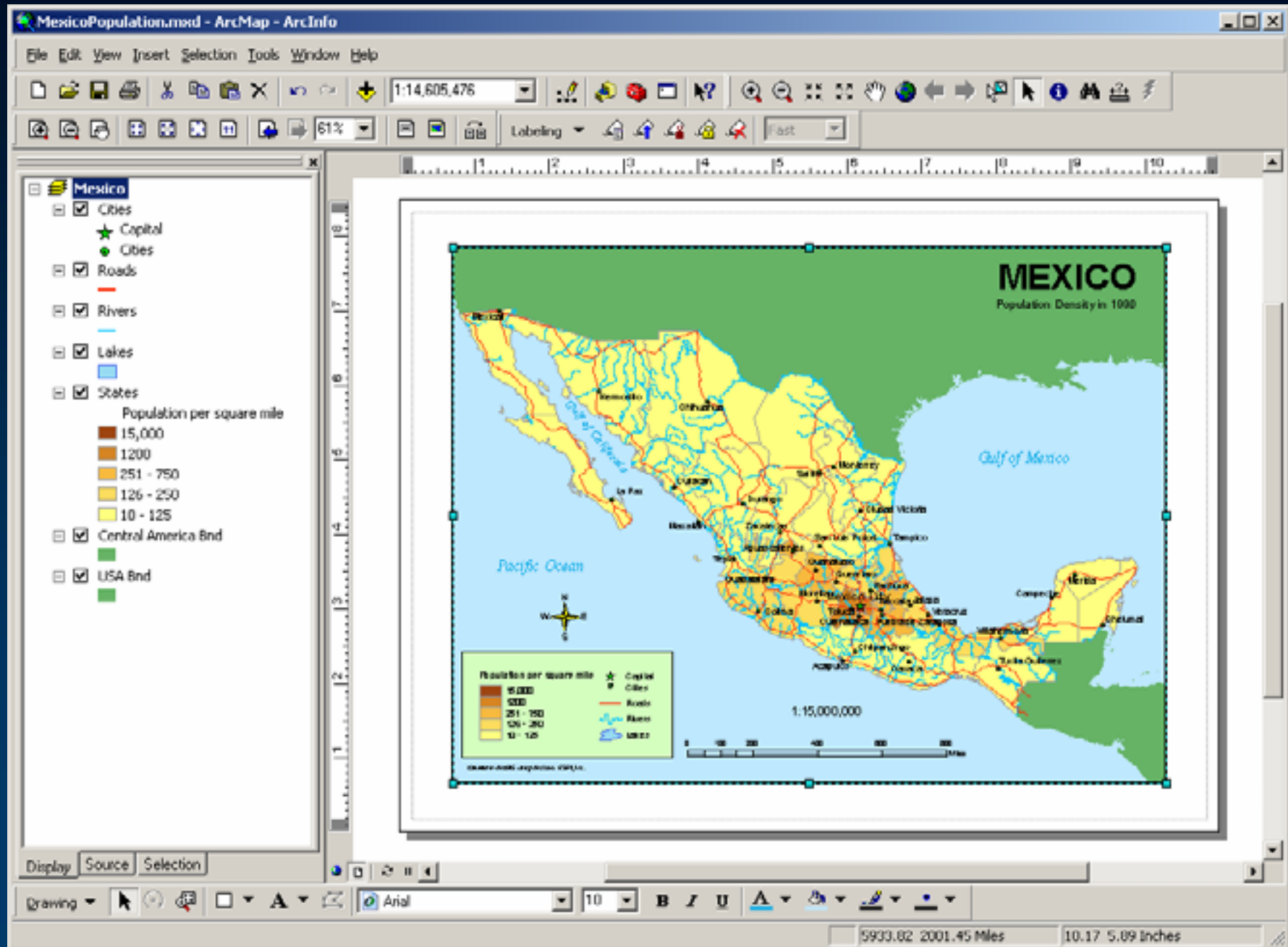


## Desktop



Desktop Extension

# Anatomy of a Map Document



# Document Views - Overview

- A map document always has at least two views, Map and PageLayout
- One or many Map objects
- One and only one PageLayout object
- View object properties and methods accessed via IActiveView
- ActiveView
  - Always only one active view (may or may not actually be activated)

# MxDocument Object

- Desktop only
- The ArcMap document
- Tightly integrated with the ArcMap application
- Commonly used in VBA
- Access via application hook in process
  - Tools, commands, dialogs
- Creation spawns a new process
- Persists as an \*.mxd

IMxDocument : IUnknown	
■—	ActivatedView: IActiveView
■□	ActiveView: IActiveView
■—	ActiveViewCommand: ICommand
■—	ContentsView (in index: Long): IContentsView
■—	ContentsViewCount: Long
■■	ContextItem: IUnknown
■□	CurrentContentsView: IContentsView
■■	CurrentLocation: IPoint
■■	DefaultColor (in Type: esriMxDefaultColorTypes): IColor
■■	DefaultTextFont: IFontDisp
■■	DefaultTextFontSize: IFontSize
—■	DelayUpdateContents: Boolean
■—	FocusMap: IMap
■—	Maps: IMaps
■—	OperationStack: IOperationStack
■□	PageLayout: IPageLayout
■■	RelativePaths: Boolean
■—	SearchTolerance: Double
■■	SearchTolerancePixels: Long
■—	SelectedItem: IUnknown
■—	SelectedLayer: ILayer
■—	StyleGallery: IStyleGallery
■—	TableProperties: ITableProperties
←	AddLayer (in Layer: ILayer)
←	CanInsertObject (pEnabled: Boolean)
←	InsertObject
←	UpdateContents

# Mapping Controls

- Engine development
- Map Control
- Page Layout Control
- Reader Control
- Symbology Control
- Add Data Dialog
- Coarse grained functions
- Fine grained access



# MapDocument

- Engine and Desktop development
- Low level file access (mxd, pmf, lyr)
- Read and/or update
- No access to extension properties / UI aspects of document
- Views are not activated when accessed

IMapDocument : IUnknown	
■	ActiveView: IActiveView
■	DocumentFilename: String
■	DocumentType: esriMapDocumentType
■	DocumentVersion: esriMapDocumentVersionInfo
■	IsMapDocument (in sDocument: String): Boolean
■	IsPasswordProtected (in sDocument: String): Boolean
■	IsPresent (in sDocument: String): Boolean
■	IsReadOnly (in sDocument: String): Boolean
■	IsRestricted (in sDocument: String): Boolean
■	Layer (in mapIndex: Long, in layerIndex: Long): ILayer
■	Map (in mapIndex: Long): IMap
■	MapCount: Long
■	PageLayout: IPageLayout
■	Printer: IPrinter
■	Thumbnail: IPicture
■	UsesRelativePaths: Boolean
←	Close
←	GetVersionInfo (versionInfoMissing: Boolean, IMajor: Long, IMinor: Long, IRevision: Long, IBuild: Long)
←	New (in sDocument: String)
←	Open (in sDocument: String, in bsPassword: String)
←	ReplaceContents (in pObject: IMxdContents)
←	Save (in bUseRelativePaths: Boolean, in bCreateThumbnail: Boolean)
←	SaveAs (in sDocument: String, in bUseRelativePaths: Boolean, in bCreateThumbnail: Boolean)
←	SetActiveView (in pActiveView: IActiveView)

# ArcGIS Server – MapServer

- Coarse grained cartographic API
- Fine grained access via IMapServerObjects
- Special behavior with respect to object pooling etc.
- ADFs

IMapServerObjects : IUnknown	
■	Layer (in MapName: String, in LayerID: Long): ILayer
■	Map (in MapName: String): IMap
■	PageLayout: IPageLayout
■	Thumbnail: IPictureDisp
←	ApplyMapDescription (in MapDescription: IMapDescription, in mapDisplay: IImageDisplay)
←	RefreshServerObjects

IMapServer : IUnknown	
■	DefaultMapName: String
■	DocumentInfo: IPropertySet
■	MapCount: Long
■	MapName (in Index: Long): String
←	ComputeDistance (in MapName: String, in fromPoint: IPoint, in toPoint: IPoint, in Units: esriUnits): Double
←	ComputeScale (in mapDesc: IMapDescription, in mapDisplay: IImageDisplay): Double
←	ExportMapImage (in mapDesc: IMapDescription, in imageDesc: IImageDescription): IMapImage
←	Find (in mapDesc: IMapDescription, in MapImage: IImageDisplay, in searchString: String, in Contains: Boolean, in searchFields: String, in option: esriFindOption, in layerIds: ILongArray): IMapServerFindResults
←	FromMapPoints (in MapDescription: IMapDescription, in mapDisplay: IImageDisplay, in mapPoints: IPointCollection, screenXValues: ILongArray, screenYValues: ILongArray)
←	GetLegendInfo (in MapName: String, in layerIds: ILongArray, in patch: IMapServerLegendPatch, in imgType: IImageType): IMapServerLegendInfos
←	GetServerInfo (in MapName: String): IMapServerInfo
←	GetSQLSyntaxInfo (in MapName: String, in LayerID: Long): ISQLSyntaxInfo
←	GetSupportedImageReturnTypes: esriImageReturnType
←	Identify (in mapDesc: IMapDescription, in MapImage: IImageDisplay, in searchShape: IGeometry, in Tolerance: Long, in option: esriIdentifyOption, in layerIds: ILongArray): IMapServerIdentifyResults
←	QueryFeatureCount (in MapName: String, in LayerID: Long, in Filter: IQueryFilter): Long
←	QueryFeatureData (in MapName: String, in LayerID: Long, in Filter: IQueryFilter): IRecordSet
←	QueryFeatureIDs (in MapName: String, in LayerID: Long, in Filter: IQueryFilter): IFIDSet
←	QueryHyperlinks (in mapDesc: IMapDescription, in MapImage: IImageDisplay, in layerIds: ILongArray): IMapServerHyperlinks
←	ToMapPoints (in MapDescription: IMapDescription, in mapDisplay: IImageDisplay, in screenXValues: ILongArray, in screenYValues: ILongArray): IPointCollection

# PageLayout

- **Container for elements**
  - **Graphic elements**
  - **Frame elements**
    - **Map frame**
  - **Surround elements**
    - **Linked to a map**
    - **Special behavior**
- **Page**
  - **Representation of logical map page**
  - **Coordinates in page space**

# Map

- **A collection of layers**
  - Methods for adding / removing / moving layers
- **Defined extents (current, full)**
- **Has a spatial reference**
- **Graphics layer built in**
- **Manages drawing**
  - Layers
  - Labeling
  - Graphics
  - Selection

# Layers

- **Presentation objects for spatial data**
  - Defines the exposed schema (rows, fields)
- **Contain drawing logic**
  - Renderers (various types depending on layer/use)
  - Advanced logic like symbol level drawing
- **Often one layer per major Geodatabase concept**
  - Feature Class – FeatureLayer
  - Topology – TopologyLayer
  - Network – Network Layer
  - etc.
- **Easily created via Layer Factory objects**
- **Common customization**
  - BaseLayer for .NET

# Feature Layer

- **Most common layer**
- **Uses a feature renderer**
- **Stores labeling properties**
- **Manages access to feature class**
  - **Selectively exposed schema**
    - **Definition Queries**
    - **Fields**
  - **Cursors**
    - **Respect layer definitions**
    - **Key concept in Geoprocessing**

# Layer types

- CadAnnotationLayer
- CadFeatureLayer
- CadLayer
- CompositeGraphicsLayer
- CoverageAnnotationLayer
- DimensionLayer
- DummyGraduatedMarkerLayer
- DummyLayer
- FDOGraphicsLayer
- FDOGraphicsSublayer
- FeatureLayer
- GdbRasterCatalogLayer
- GraphicsSubLayer
- GroupLayer
- IMSMapLayer
- MapServerLayer
- NetworkLayer
- RasterCatalogLayer
- RasterLayer
- TinLayer
- Topology

## • Interfaces

### – ILayer

### – IGeoFeatureLayer

- FeatureLayer
- CadFeatureLayer
- GdbRasterCatalogLayer

### – IRasterLayer

- RasterLayer

### – ICompositeLayer

- IAnnotationLayer
  - FDOGraphicsLayer

### – IGeodataset

# Renderers

- **Renderers define symbols for features**
  - Perform the drawing
- **Feature Renderers**
  - Different renderers for different classifications situations
  - Special attribute driven renderers
    - Chart, Dot Density, Representation
- **Raster Renderers**
  - Classification
  - Special raster specific situations
- **Tin Renderers**
  - Face, Breakline etc
- **Network Renderer**
- **Common customization**

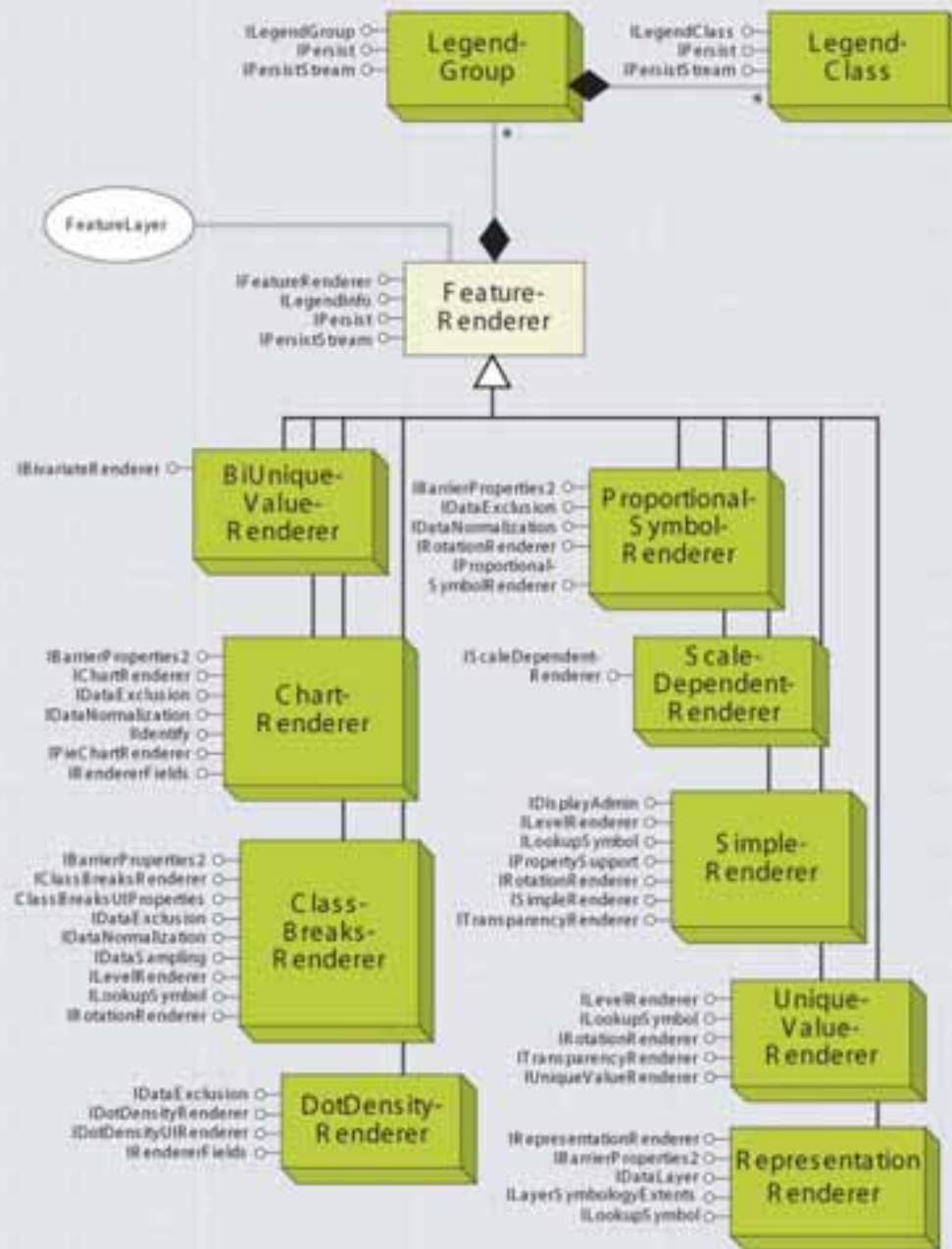
# Representations

- **Representations are:**
  - An intelligent way to symbolize features
  - A storage model that stores symbol information in the geodatabase for re-use and sharing
    - **Properties of a feature class**
      - Feature classes can have more than one representation
  - Managed through the representation renderer
  - Distinct symbology model
- **Representations are NOT:**
  - A new data type
  - A new layer type

# Representations (continued)

- **RepresentationWorkspaceExtension**
  - Key GeoDatabase component for managing representations
  - Methods for creating and opening feature class representations
- **RepresentationClass**
  - Represents a specific feature class representations
  - Provides access to override properties
    - Override fields
    - Whether or not to override shape
  - Provides access to representation rules
    - Collection of symbol graphic primitives
    - Enhanced capabilities via Geometric effects
      - Key developer customization

# Feature Renderer Objects

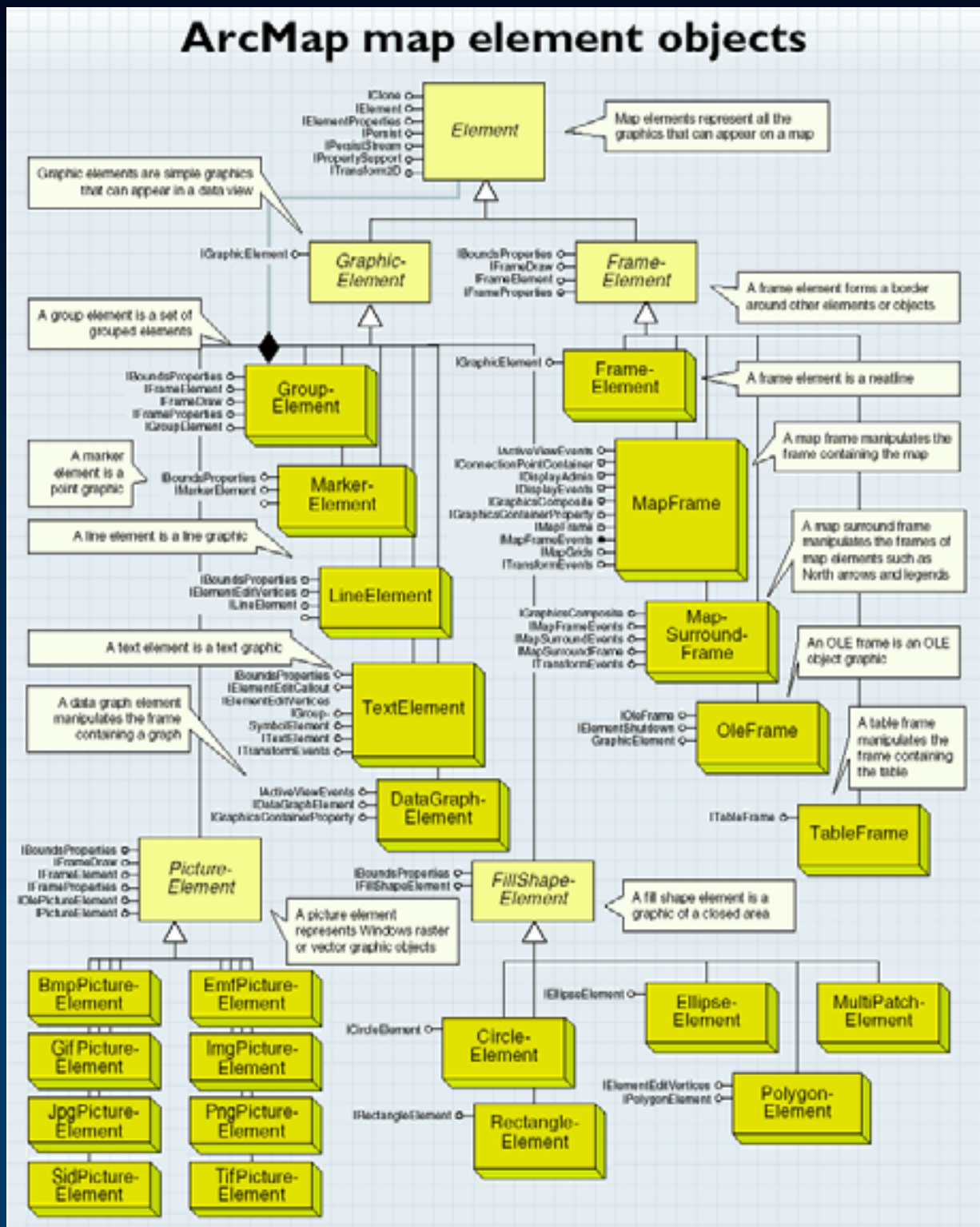


# Labeling

- **Different than feature drawing**
  - Drawn by the map
  - Map has a specific label engine
- **Layers define labeling properties**
  - Placement logic relative to geometry
  - Text symbols
  - Mini logic in label expressions to create label text
- **Label engine processes properties and features**
  - Produces optimal placement
  - Choice of two label engines
    - Standard Label Engine
    - Maplex Label Engine – now available for Engine and Server
- **Conversion to annotation**
  - Allows for editing
  - Coarse grained object to perform this task

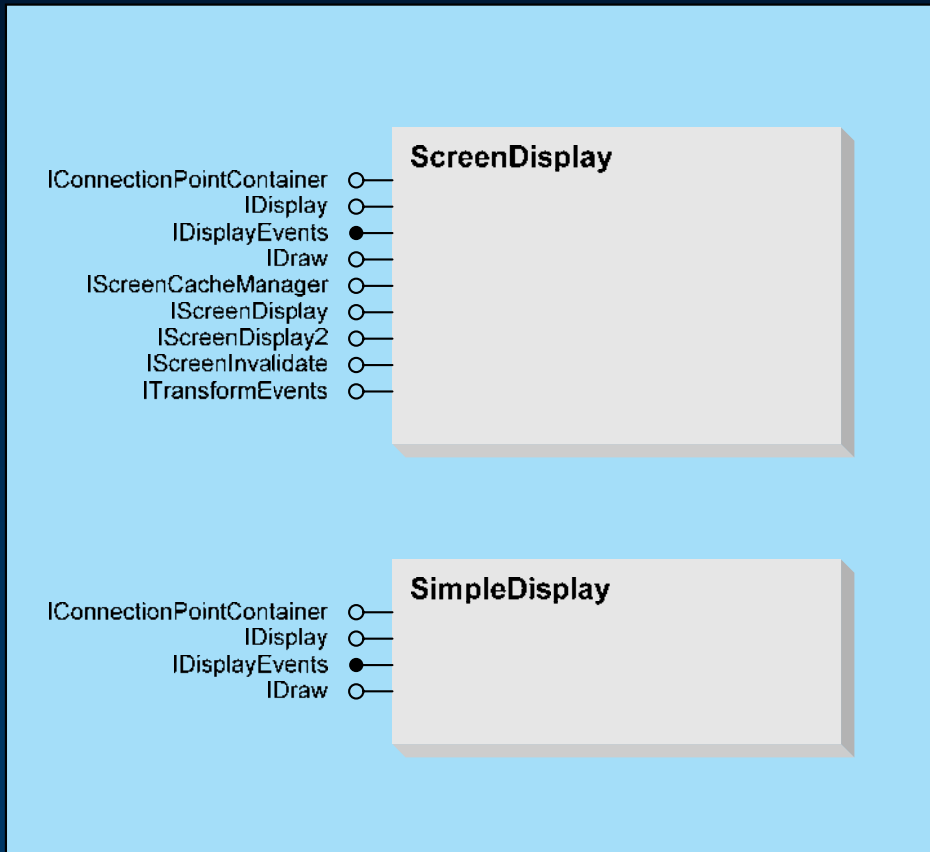
# Elements

- Various element types for different purposes
  - Graphic elements (map or page)
  - Frame elements (page)
- Frame elements often link to larger concepts
  - Map
  - Legend
  - Grids/Graticules



# Displays - Overview

- Every view has a **ScreenDisplay**
- Other Display objects can be created for special use
  - Example: the **IActiveView::Output** method creates and uses its own **SimpleDisplay** object to draw during printing or exporting.



# ArcObjects Display and Windows GDI

- ArcObjects drawing internally uses Windows GDI (Graphical Device Interface)
- GDI is part of the Windows operating system (Win32)
- GDI is documented as part of the Windows Platform SDK. Available online at [msdn.microsoft.com](http://msdn.microsoft.com).
- Display properties and methods expose and use many GDI constructs:
  - hWnd
  - hDC

# Display – DisplayTransformation

- Each Display object has one **DisplayTransformation**
  - defines how real-world coordinates are mapped to a output device
  - container for all bounds and extent information for the view (Map or PageLayout)
  - helper methods for transforming coordinates between world space to device space
    - world space – geographic coordinates
    - device space – pixels

IDisplayTransformation : ITransformation	
■	Bounds: IEnvelope
■	ConstrainedBounds: IEnvelope
■	DeviceFrame: tagRECT
■	FittedBounds: IEnvelope
■	ReferenceScale: Double
■	Resolution: Double
■	Rotation: Double
■	ScaleRatio: Double
■	SpatialReference: ISpatialReference
■	SuppressEvents: Boolean
■	Units: esriUnits
■	VisibleBounds: IEnvelope
■	ZoomResolution: Boolean
←	FromMapPoint (in mapPoint: IPoint, out x: Long, out y: Long)
←	FromPoints (in pointDistance: Double): Double
←	ToMapPoint (in x: Long, in y: Long): IPoint
←	ToPoints (in mapDistance: Double): Double
←	TransformCoords (in mapPoints: WKSPoint, in devPoints: tagPOINT, in numPoints: Long, in options: Long)
←	TransformRect (in mapRect: IEnvelope, in devRect: tagRECT, in options: Long)

# Animations

- Map supports animation capabilities via the Animation library
  - Animation property and utility classes
  - Export animations to video formats
  - 2D animation capabilities for Desktop only

## Time Layer Animation



Data provided courtesy of Declan Butler <http://declanbutler.info/blog>

# Dynamic Display

- **High Performance Drawing in 2D Maps**
- **Uses OpenGL to Perform Drawing**
  - No Knowledge of OpenGL Required
  - ArcObjects API for Drawing to Dynamic Display
  - OpenGL context accessible
- **Active Display**
- **Integrated with and using ArcObjects**
- **Developers Create 'Dynamic' Layers**
- **Map handles the Drawing of Dynamic Layers**
- **Display content with high refresh rates**
- **Smooth panning, rotation and zooming**
- **Using cartographically rich maps**

# Output - Overview

- **ArcObjects for Printing or Exporting**
- **Like Display, based on Windows GDI**
  - large output images always use a temporary file on disk (Windows Enhanced Metafile - EMF)
- **Each output object is a converter that may:**
  - prepare and spool the Display's GDI instructions for conversion by a printer driver
  - convert the Display's GDI instructions into PDF, PostScript, or SVG draw instructions
  - rasterize the GDI instructions into a raster image saved as BMP, PNG, JPEG, TIFF, or GIF

# Output - Printing



- **Printer objects connect the application to the native printer drivers installed on the system**
- **EmfPrinter**
  - the most common way to print
  - picks up the EMF file after drawing is complete and feeds it to the printer manufacturer's driver for conversion
- **PsPrinter**
  - application-level driver
  - only recommended when color separated files are needed
- **ArcPressPrinter**
  - application-level driver
  - Desktop only
  - OutputExtensions assembly

# Output - Printing

IPaper2 : IUnknown	
■	Resolution: Integer

IPaper : IUnknown	
■	FormID: Integer
■	FormName: String
■	Forms: IEnumNamedID
■	Orientation: Integer
■	PrintableBounds: IEnvelope
■	PrinterInfo (out hDevMode: OLE_HANDLE): OLE_HANDLE
■	PrinterName: String
■	TrayID: Integer
■	Trays: IEnumNamedID
■	Units: esriUnits
←	Attach (in hDevMode: OLE_HANDLE, in hDevNames: OLE_HANDLE)
←	QueryPaperSize (out Width: Double, out Height: Double)

IPrinter : IUnknown	
■	DriverName: String
■	FileExtension: String
■	Filter: String
■	Name: String
■	Paper: IPaper
■	PrintableBounds: IEnvelope
■	PrintToFile: String
■	Resolution: Integer
■	SpoolFileName: String
■	StepProgressor: IStepProgressor
■	Units: esriUnits
←	DoesDriverSupportPrinter (in PrinterName: String): Boolean
←	FinishPrinting
←	QueryPaperSize (out Width: Double, out Height: Double)
←	StartPrinting (in PixelBounds: IEnvelope, in hDcPrinter: OLE_HANDLE): OLE_HANDLE
←	VerifyDriverSettings: Boolean

- The Paper object works together with one of the Printer objects to control printing
- IPaper interface PrinterName property determines which print driver the application will print to
- IPrinter interface has a Paper property which must be set for the print output to be correct
- IPaper interface
  - Persisted in the document when IPage::FormID = esriPageFormSameAsPrinter
  - The IPaper interface is a wrapper around the GDI DEVMODE and DEVNAME structures.
- Refer to the Print Active View, ModifyMapPageAndPrinter sample

# Output - Export

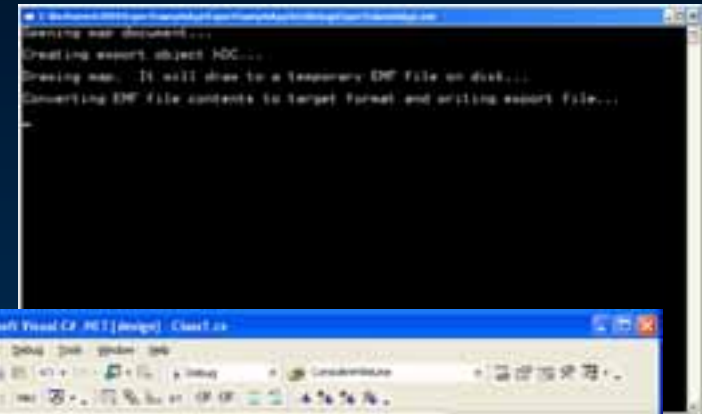
- Ten export objects for ten export file formats
  - Raster: BMP, JPEG, PNG, TIFF, GIF
  - Vector: EMF, PDF, EPS, Illustrator, SVG
- Basic object setup and code for all ten objects is identical
- Refer to the Export Active View sample
- Objects are the same ones used to generate ArcGIS Server images

IExport : IUnknown	
■	ExportFileName: String
■	Filter: String
■	Name: String
■	PixelBounds: IEnvelope
■	Priority: Long
■	Resolution: Double
—□	StepProgressor: IStepProgressor
—□	TrackCancel: ITrackCancel
←	Cleanup
←	FinishExporting
←	StartExporting: OLE_HANDLE

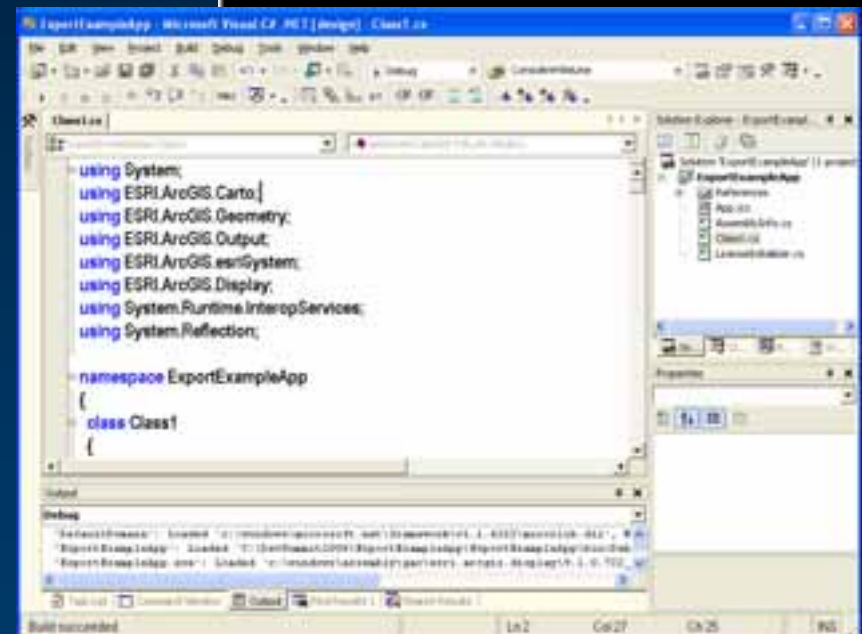
# Sample Application

## Open a Map Document and Export to PDF

- Valid for all product types and license levels
- Simple C# command line application
- Objects from five ESRI assemblies:
  - Carto
  - Display
  - Output
  - Geometry
  - System
- Essentially two samples and five new lines of code



```
Opening map document...
Creating export object AOC...
Drawing map. It will show to a temporary PDF file on disk...
Converting PDF file contents to target format and writing export file...
```

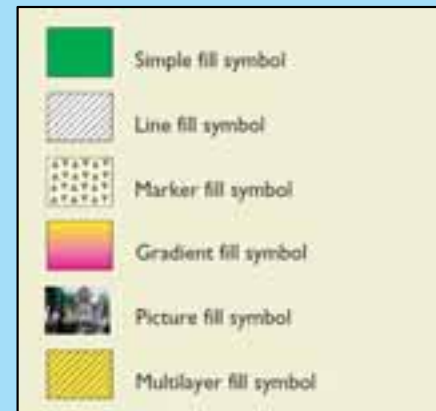
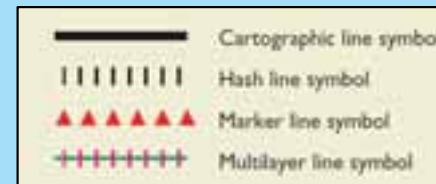
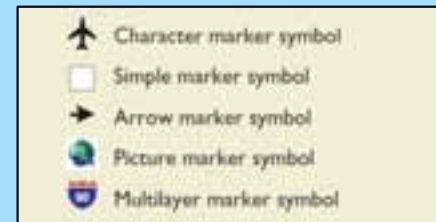


```
using System;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Output;
using ESRI.ArcGIS.esriSystem;
using ESRI.ArcGIS.Display;
using System.Runtime.InteropServices;
using System.Reflection;

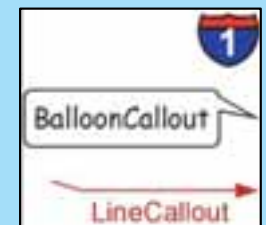
namespace ExportExampleApp
{
    class Class1
    {
    }
}
```

# Symbols

- Symbols are the elemental graphical object
- Symbols contain the logic that draws with GDI
- Basic types
  - Marker Symbol
    - symbolizes point geometries
  - Line Symbol
    - symbolizes line geometries
  - Fill Symbol
    - symbolizes polygon geometries
  - Text Symbol
    - handles special properties of text

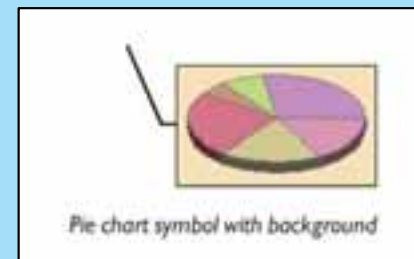
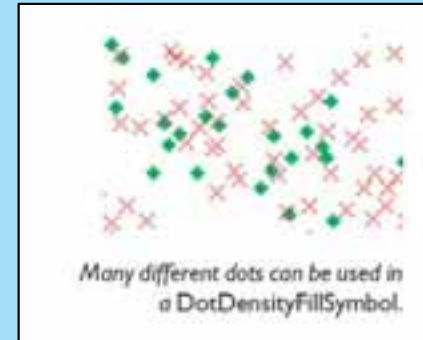


It was a dark and stormy night



# Symbols

- **Other symbol types (data driven)**
  - **3D Pie Chart symbol**
    - uses a pie chart renderer
  - **Dot Density Fill Symbol**
    - uses a dot density renderer
  - **Representation symbology**
    - Can be used only with Representation Classes



# Styles

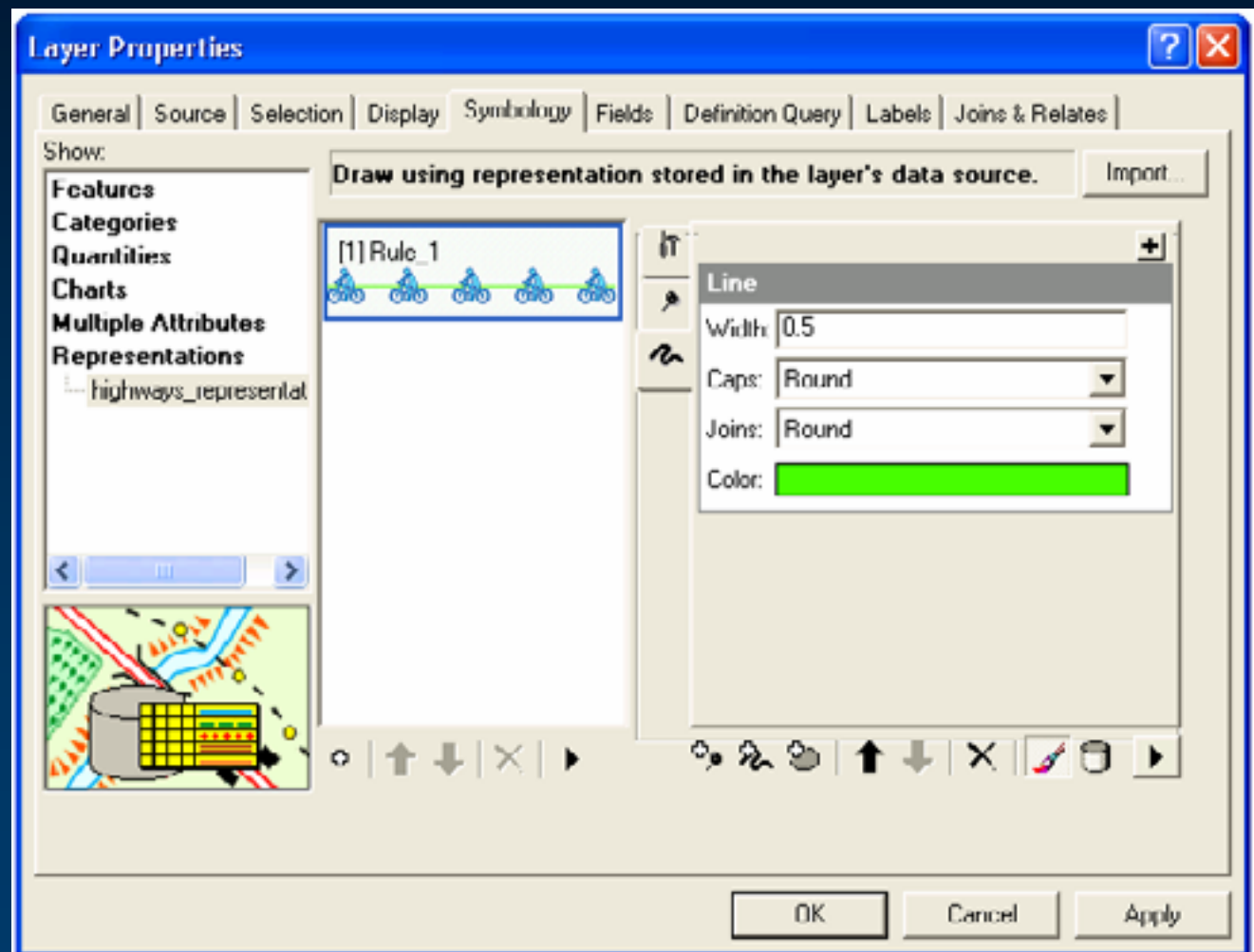
- **Styles are collections of symbols and map elements**
  - Desktop styles
  - Server styles (Engine and Server)
    - Server styles are generated via a utility
- **Access via StyleGallery or ServerStyleGallery**
  - Style gallery contains StyleGalleryClasses
    - Access styles of a specific type via its StyleGalleryClass
  - StyleGalleryItems
    - Persist the symbol as an item in the style file

# Recommended customizations

- **First, check provided objects**
- **Consider a custom UI that uses existing objects**
  - Tools, commands, factories
- **If not, then write a custom object**
  - Geometric Effects
  - Renderers
  - Layers
  - Elements
  - Symbols
- **Follow existing patterns in ArcGIS**
- **EDN**
  - Working with ArcGIS Components: Mapping and Visualization

# Walkthrough 2: Custom Geometric Effect

- Custom geometric effect (C#)
- Plugs into standard UI



# Related Sessions and Evaluations Reminder

## Today

- **Cartography API Tech Talk**
  - after this session, Community Center Tech Talk Area 6

## Wednesday

- **ArcGIS Engine and ArcGIS Desktop Panel Discussion**
  - 2:45 p.m. to 4 p.m., Primrose B
- **Meet the Teams – Mapping and Visualization Team**
  - 4 p.m., Oasis 4

## Thursday

- **Developer's Guide to 3D Visualization in ArcGIS 9.2**
  - 10:15 a.m. to 11:30 a.m., Catalina/Madera

**Session Attendees:**

**Please turn in your session evaluations.**

***... Thank you***



Questions?