Intro to MapObjects

February 7, 2003
Mark McDonald

What is (are) MapObjects?

- MapObjects-Windows Edition
  - ActiveX control with ca. 50 programmable objects
  - Suitable for various development environments:
    - Visual Basic
    - Visual C++
    - Visual FoxPro
    - . . .
  - Current VISC version (2.1) predates .NET
  - Jan 2003 release of version 2.2
- MapObjects LT
  - Basic functionality
- MapObjects- Java Standard Edition
  - Recently released (early 2002)
  - Completely different implementation (Java Beans)
  - For multi-platform use

Differences with other ESRI Products

- ArcGIS Desktop "Geographic data creation, management, integration, analysis"
  - Common interface
    - ArcMap
    - ArcCatalog
    - ArcToolbox
  - Two products with common interface
    - ArcView 8.x
    - ArcInfo – more advanced and comprehensive
- Developer Tools
  - ArcObjects
    - "The technology framework of ArcGIS"
    - Customizes, extends: ArcInfo, ArcEditor, ArcView 8.x
    - Requires one of above
  - MapObjects
    - Adapted for "foreign" (non-ESRI) components
    - Suitable for more general use

Intro to Map Objects

This presentation assumes:

- Basic knowledge of ESRI products
  - ArcView, or
  - ArcInfo
- Basic programming skills
  - MS Visual Basic 6, or
  - other programming language

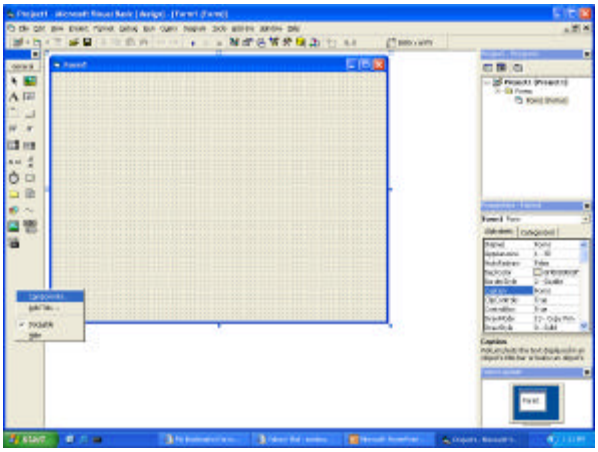
Objectives:

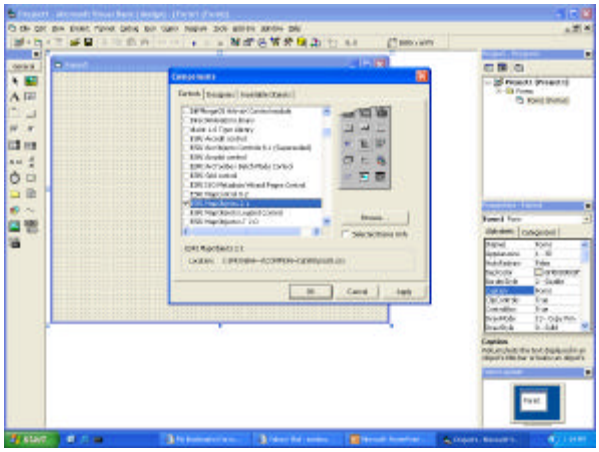
- Know what's possible
- Reference on where to start

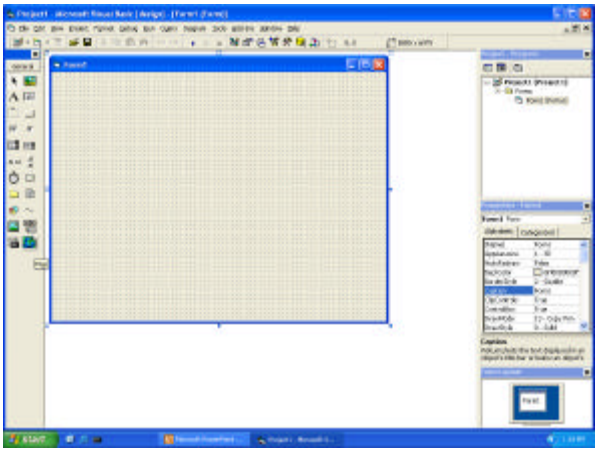
Basics

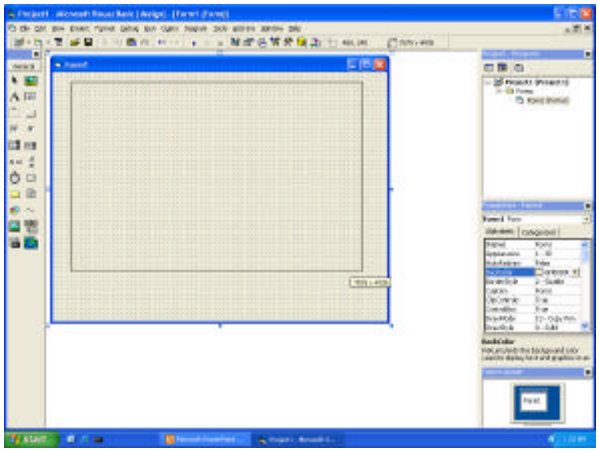
- Creating a Map
  - Add MapObjects Control to VB Toolbox
  - Add Map to a form
  - Add data layer(s) to Map

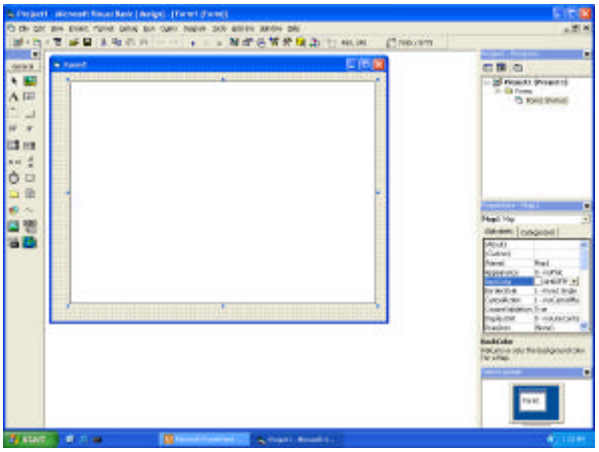
Project1.vbp

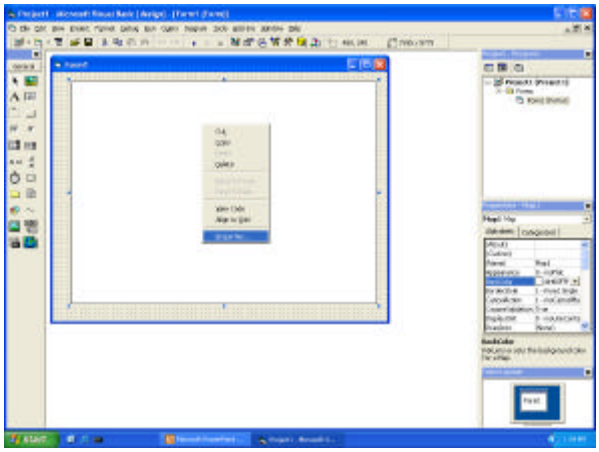


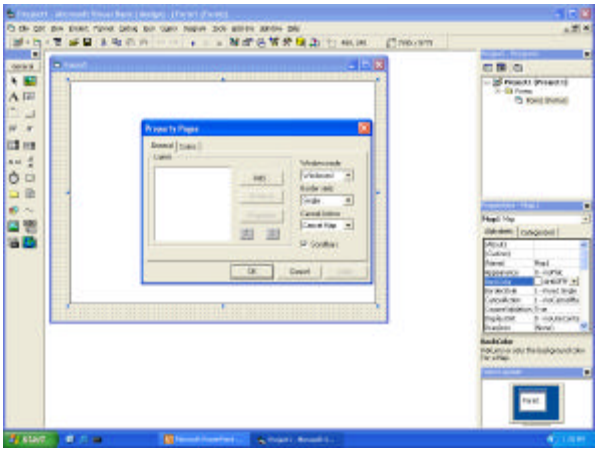


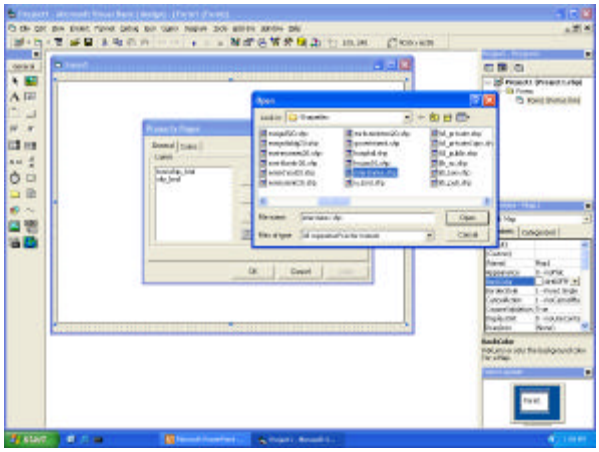


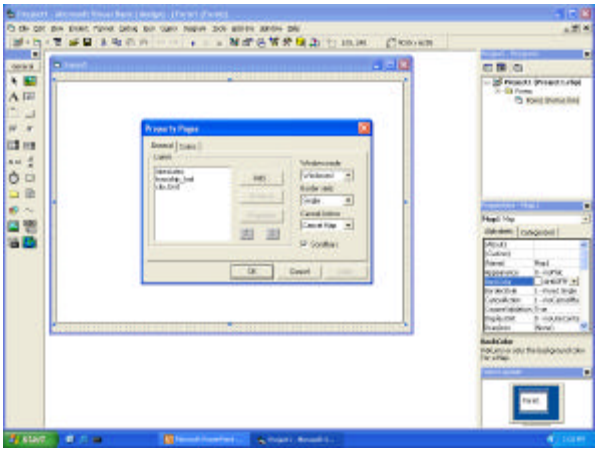


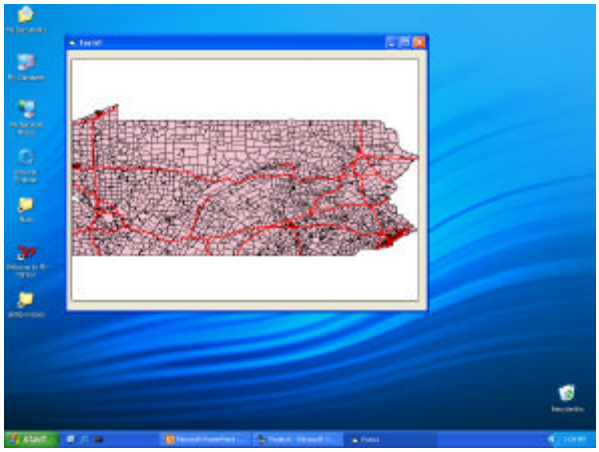








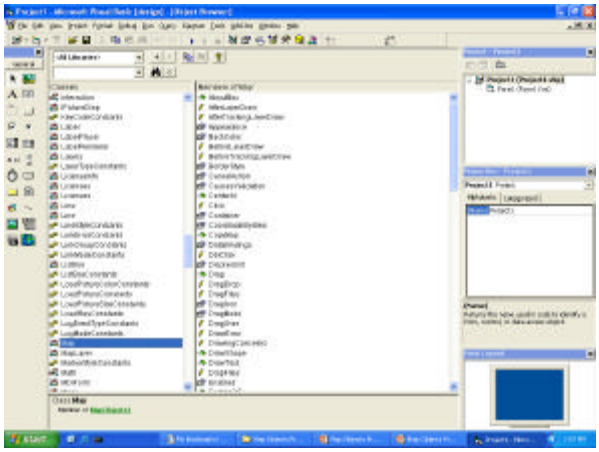


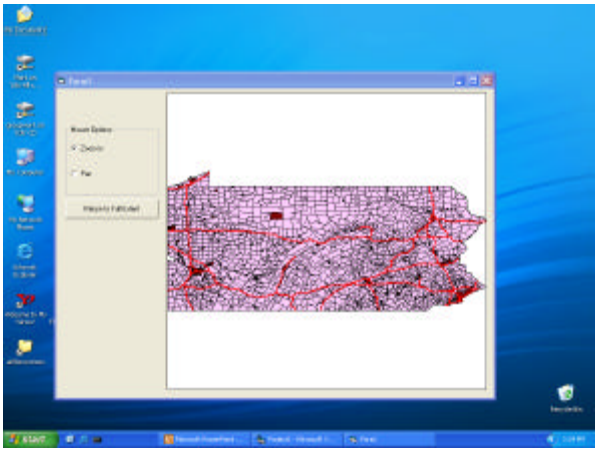


MapObjects Basics

- Basics
  - Add MapObjects Control to VB Toolbox
  - Add Map to a form
  - Add data layer(s) to Map
- Map manipulation
  - Add VB controls
  - Add three lines of code
  - Results in ability to:
    - Zoom In
    - Pan
    - Zoom Out to original view

Draw VB controls on blank form
Object Browser view
Project2.vbp





```
Private Sub Command1_Click()
    'Extent:  Returns or sets the spatial extent of an object
    'FullExtent: Returns or sets a special Rectangle object
    '                  (Rectangle represents the bounding box of a Map)
    Map1.Extent = Map1.FullExtent
End Sub


Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, _
          X As Single, Y As Single)

    If Option1.Value = True Then ' Zoom In
        'Function TrackRectangle():
        '   Rubber-bands a rectangle on the Map and returns a Rectangle object
        Map1.Extent = Map1.TrackRectangle

    ElseIf Option2.Value = True Then ' Pan
        'Sub Pan(): Tracks the mouse while panning the Map
        Map1.Pan
    End If

End Sub
```

Map Objects by Type

- Data Access
  - Obtain geographic data from database
- Projection
  - Transform 3-D data for display
- Map Display
  - Display data on a 2-D map
- Address Matching
  - Match address(es) with location(s) on map
- Geometric & Utility
  - Create and manipulate points, lines, polygons, ...
  - Manipulate text strings
- MapObjects Helper Components
  - Add Legend, Scalebar

Comments:
- objects grouped by function
- not an object hierarchy

Map Objects

- Data Access
  - **DataConnection**
    - Represents connection to geographic data source
    - Analogous to VB Connection object
  - **GeoDataset**
    - Represents layer of geographic data on a map
    - Associated with a MapLayer object
    - Can be in most vector data formats
  - **Recordset**
    - Represents records from a GeoDataset, or ….
    - ….resulting from a query
  - **Statistics**
    - Result of calculation on a numeric Field of a Recordset using Recordset object's CalculateStatistics method
  - **Other objects- . . .**
- Projection
- Map Display
- Address Matching
- Geometric & Utility
- MapObjects Helper Components

Map Objects

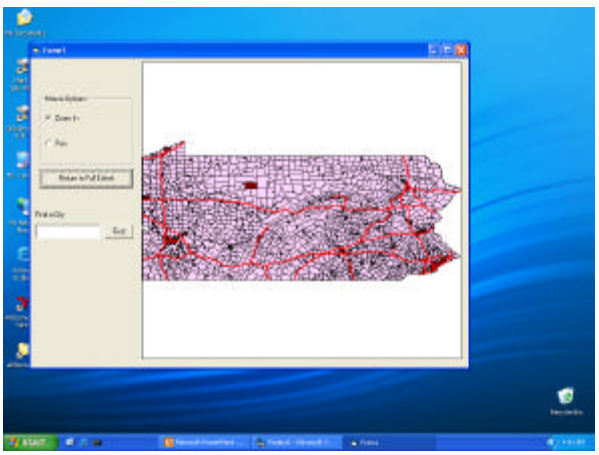
- Data Access
  - DataConnection
  - GeoDataset (GeoDatasets Collection)
  - Recordset
  - **Other objects**
    - TableDesc
      - Represents a description of the Fields collection of a Recordset.
      - Return or set properties for Field in a Recordset, where appropriate, i.e. within the limits of the GeoDataset from which the Recordset is derived
      - FieldLength, FieldName, FieldPrecision, FieldScale, FieldType among choices
    - Table
      - read-only object corresponding to a table in a relational database
      - Used for standard relational DB's, not geodatasets
    - Field (Fields Collection)
      - represents a column of data within a Recordset with a common data type and a common set of properties.
- Projection
- Map Display
- Address Matching
- Geometric & Utility
- MapObjects Helper Components

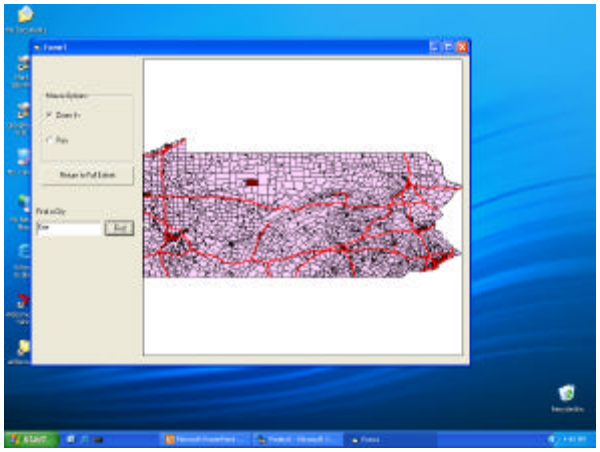
Searching GeoDatasets

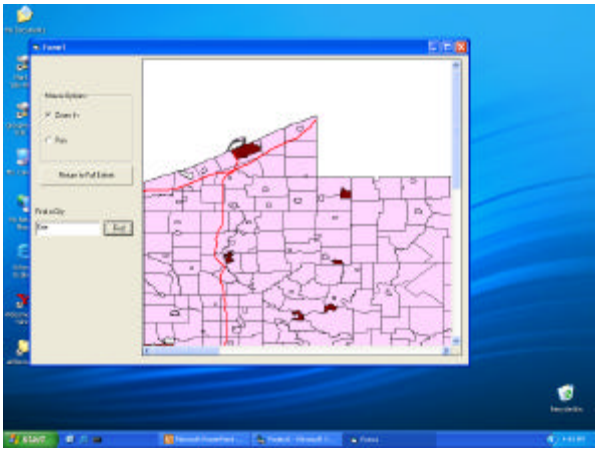
Can be done by searching for:

- Text in an attribute
- Proximity to a spatial feature

Project3.vbp







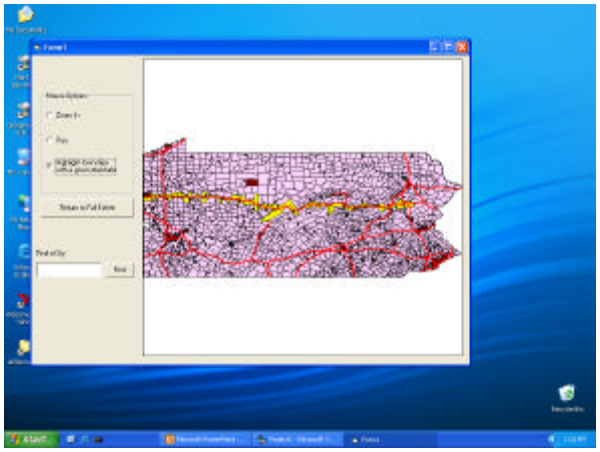
```
Private Sub Command2_Click()
    ' build a search expression
    Dim exp As String
    ' "NAME" is field name in shapefile city_bnd.shp
    exp = "NAME = '" & UCase(Text1.Text) & "'"

    ' perform the search
    Dim recs As MapObjects2.Recordset
    Set recs = Map1.Layers("city_bnd").SearchExpression(exp)

    ' show the results, if any
    If Not recs.EOF Then
       Dim shp As Object
       Set shp = recs.Fields("Shape").Value
       Dim rect As MapObjects2.Rectangle

       Set rect = shp.Extent     ' rect equals city's extent
       rect.ScaleRectangle 10    ' Scale up rect by factor of 10
       Set Map1.Extent = rect    ' Make the map's extent = rect
       Map1.Refresh              ' redraw the map
       Map1.FlashShape shp, 10 ' flash it so we see it
    End If
End Sub
```

Searching GeoDatasets

Can be done based by searching for:

- Text in an attribute
- Proximity to a spatial feature

Project3.vbp



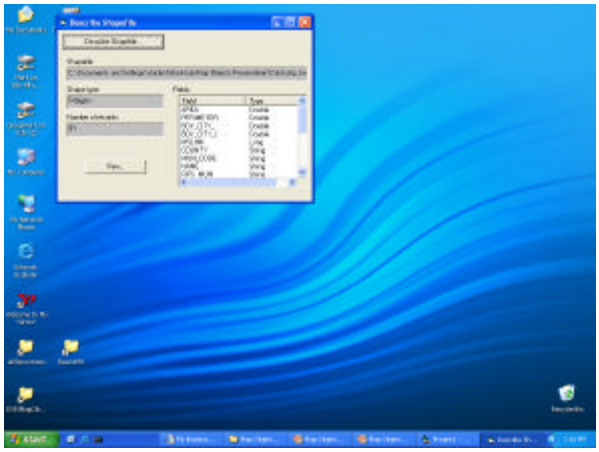
```
Dim gSel As MapObjects2.Recordset
   .
   .
   .
Private Sub Map1_MouseDown(Button As Integer, ……
            ……..
    ElseIf Option3.Value = True Then
        ' spatial query
        Dim p As Point
        Set p = Map1.ToMapPoint(x, y) 'Converts computer's (x,y) to map's (x,y)

        ' search for a highway within the tolerance
        Dim recs As MapObjects2.Recordset
        Set recs = Map1.Layers("interstates"). SearchByDistance(p, Map1. _
             ToMapDistance(200), "")
        'SearchByDistance() searches "interstates.shp" for anything near Point p
         'ToMapDistance() converts computer distance units to map distance units

        ' If nothing is found
        If recs.EOF Then
            Set gSel = Nothing
        ' Else search for townships intersecting
        Else
            Set gSel = Map1.Layers("township_bnd") _
                .SearchShape( recs.Fields ("Shape" ).Value, moAreaIntersect, "")
        End If

        Map1.Refresh ' trigger a redraw of the map
    End If
```

```
Private Sub Map1_AfterLayerDraw(ByVal index As _
    Integer, ByVal canceled As Boolean, ByVal hDC _
    As stdole.OLE_HANDLE)

    If Map1.Layers(index).Name = "township_bnd" Then

        If Not gSel Is Nothing Then
            Dim sym As New MapObjects2.Symbol
            sym.Color = moYellow
            Map1.DrawShape gSel, sym
        End If

    End If

End Sub
```

Connecting to a GeoDatabase

- Data not known during development?
- Connect to data as needed!
- Allows flexibility

DescribeShapefile.Vbp



```
'This example demonstrates using a
'MapLayer object without attaching it to a Map control.
Dim g_layer As MapLayer

Private Sub Command1_Click()
'Use Common Dialog Form to select a particular shapefile
  CommonDialog1.Filter = "ESRI Shapefiles (*.shp)|*.shp"
  CommonDialog1.ShowOpen
  If Len(CommonDialog1.FileName) = 0 Then Exit Sub
  . . . . .
  Dim dc As New DataConnection
  dc.Database = CurDir ' CurDir = VB function returning String of current directory

  If Not dc.Connect Then Exit Sub
   . . . . .
  Dim name As String
  name = Left(CommonDialog1.FileTitle, Len(CommonDialog1.FileTitle) -4) 'Remove ".shp"
  Dim gs As GeoDataset
  Set gs = dc.FindGeoDataset(name)  'Get geodataset

If gs Is Nothing Then Exit Sub
  ' Make a new MapLayer based on this new GeoDataset
  Set g_layer = New MapLayer
  Set g_layer.GeoDataset = gs
  . . . .
```

Map Objects by Type

- Data Access
- Projection
  - **Datum**
    - Defines datum, the basis of geographic coordinate system
  - **GeoCoordSys**
    - Represents a geographic coordinate system (GCS)
    - Positions described with latitude-longitude (degrees on world surface)
    - View: a *3-D spheroid*
  - **ProjCoordSys**
    - Represents a projected coordinate system
    - Positions described with X and Y coordinates on map
    - View: a *2-D map*
  - **Other objects . . . .**

--- Typically use ESRI-predefined constants for Datum, GCS, PCS ---

- Map Display
- Address Matching
- Geometric & Utility
- MapObjects Helper Components

Map Objects by Type

- Data Access
- Projection
  - Datum
  - GeoCoordSys
  - ProjCoordSys
  - **Other objects:**
    - Projection- Specifies mathematical transformation of GCS to projected coordinates
    - Spheroid - specified from pre-defined SpheroidConstants or by user-definition
    - PrimeMeridian- defines line of zero longitude for coordinates in a GCS/GeoCoordSys object
    - Unit- defines units of measurement used in a GeoCoordSys or ProjCoordSys object.
    - GeoTransformation - object converts vector data from one coordinate system to another (geographic transformation or datum shift)
- Map Display
- Address Matching
- Geometric & Utility
- MapObjects Helper Components

Map Objects by Type

- Data Access
- Projection
- Map Display
- Address Matching
- Geometric & Utility
- MapObjects Helper Components

Map Display Objects

- **Map**- A Map control displays a collection of Layers.
- **Layers Collection**- Layers based on geographic data:
  - **MapLayer**
    - Vector geographic data
    - Represents a GeoDataset data layer on a Map
    - Can be from ESRI shapefile, an SDE layer, an ARC/INFO coverage, CAD files and VPF data.
  - **ImageLayer**
    - Raster data from an image file
    - Has additional transformation data (allows alignment with vector MapLayer objects)
  - **TrackingLayer**
    - Displays geographic phenomena that may change position
    - Represented as GeoEvent objects
- **Symbol**- control how a feature or shape is displayed
- . . . . . .

Map Display Objects: *Renderers*

Renderers- Symbolize feature(s) of a MapLayer:

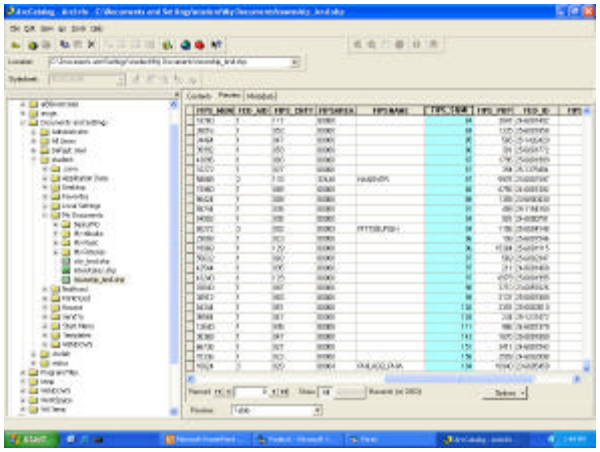
- *ChartRenderer*
  - Depict attribute values as elements of a single pie or bar chart, or
  - Show multiple charts, varying relative size of each feature's chart
- *ClassBreaksRenderer*
  - Break dataset into categories or classes based on some specific numeric attribute
  - Draw different symbol for each class
- *DotDensityRenderer*
  - draw dots on a feature, dot density proportional to a feature's value
  - e.g., dot density proportional to population density
- *ValueMapRenderer*
  - draw a Symbol for each unique data value
- *ZRenderer*
  - symbolize Z values of features in a MapLayer
  - (Z: third-dimension, perpendicular to X & Y values in map plane)
- *GroupRenderer*
  - associate multiple renderers with a given MapLayer

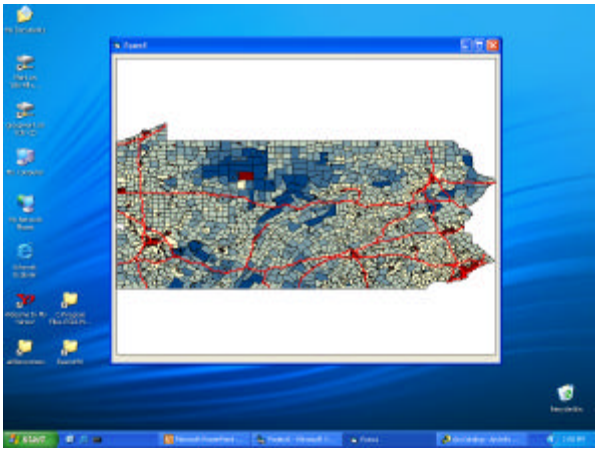
MapObjects

Using ClassBreaksRenderer

ArcCatalog: township_bnd.shp

Project4.vbp





```
Private Sub Form_Load()
    ' counties layer
    Dim rC As New MapObjects2.ClassBreaksRenderer
    Set Map1.Layers("township_bnd").Renderer = rC
    rC.Field = "FIPS_SQMI"

    Dim stats As MapObjects2.Statistics
    Set stats = Map1.Layers("township_bnd").Records.CalculateStatistics("FIPS_SQMI")

    ' calculate breaks away from the mean,
    ' only add breaks within the range of values
    Dim breakVal As Double
    breakVal = stats.Mean - (stats.StdDev * 3)

    Dim i As Integer
    For i = 0 To 6
        If breakVal >= stats.Min And breakVal <= stats.Max Then
            rC.BreakCount = rC.BreakCount + 1
            rC.Break(rC.BreakCount - 1) = breakVal
        End If
        breakVal = breakVal + stats.StdDev
    Next i

    rC.RampColors moPaleYellow, moNavy
End Sub
```

Map Display Objects: *Labeling*

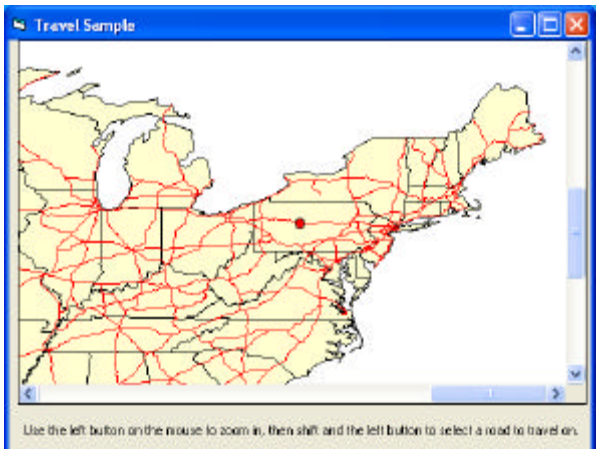
- *LabelRenderer* - draw label on a feature.
- *TextSymbol* - attributes that control how text is rendered
- *LabelPlacer* - similar to LabelRenderer but finer adjustments possible

Map Display Objects: *Events*

- GeoEvent
  - Phenomenon whose geographic position may change
  - Displayed on a TrackingLayer
- EventRenderer
  - Renders GeoEvent(s) occurring on a MapLayer
  - Draws a Symbol for each event

Tracking a GeoEvent

- Use for objects which may move or be moved




```
' User selects highway with SHIFT+ Mouse Click
' GeoEvent moves along the length of that highway segment

Dim gLine As Object       ' gLine:  a line object that gets extracted from the highways
Dim gCurPoint As Integer  ' gCurPoint:  keeps track of current point in gLine
. . . . .
. . . . .
Private Sub Form_Load()
  ' load data into the map
   Dim dc As New DataConnection
   dc.Database = ReturnDataPath("usa") ' ReturnDataPath not shown- simply returns data directory
   If Not dc.Connect Then End

   Dim layer As MapLayer
   Set layer = New MapLayer
   Set layer.GeoDataset = dc.FindGeoDataset("states")
   layer.Symbol.Color = moPaleYellow
   Map1.Layers.Add layer

   Set layer = New MapLayer
   Set layer.GeoDataset = dc.FindGeoDataset("ushigh")
   layer.Symbol.Color = moRed
   Map1.Layers.Add layer

   ' set the symbol of the TrackingLayer
   Map1.TrackingLayer.Symbol(0).Style = moCircleMarker
   Map1.TrackingLayer.Symbol(0).Color = moRed
   Map1.TrackingLayer.Symbol(0).Size = 6
End Sub
```

```
Sub DoTravel(x As Single, y As Single)
' convert the selected point to map coordinates & search for a highway
Set pt = Map1.ToMapPoint(x, y)
Set recs = Map1.Layers(0).SearchByDistance(pt, Map1.ToMapDistance(100), "")
 ' if a highway is found, extract the shape  and store it in gLine
 If Not recs.EOF Then
   Set gLine = recs("Shape").Value
   Set pt = gLine.Parts(0).Item(0)  ' get first point
   gCurPoint = 0              ' initialize the point counter

   ' add an event
   Map1.TrackingLayer.ClearEvents
   Map1.TrackingLayer.AddEvent pt, sym
   Timer1.Interval = 75
 End If
End Sub

Private Sub Timer1_Timer()
 ' if the point counter reaches the end of the line
 ' reset the counter, the timer, and the TrackingLayer
 If gCurPoint = gLine.Parts(0).Count - 1 Then
   gCurPoint = 0
   Timer1.Interval = 0
   Map1.TrackingLayer.ClearEvents
 Else
   ' move the event to the next position along the line
   ' then increment the point counter
   Set pt = gLine.Parts(0)
   Map1.TrackingLayer.Event(0).MoveTo pt(gCurPoint).x, pt(gCurPoint).y
   gCurPoint = gCurPoint + 1
 End If
End Sub
```

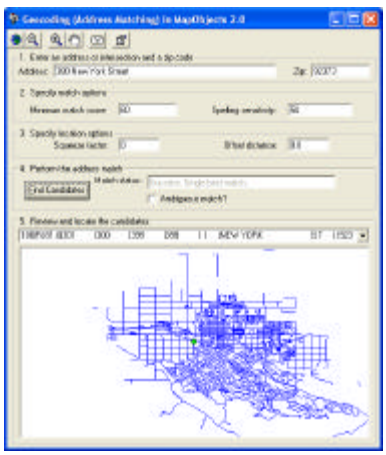
Map Objects by Type

- Data Access
- Projection
- Map Display
- Address Matching
  - Geocoder - allows specification of address, street intersection, or table of addresses to match against a street network
  - AddressLocation - represents the results of an address match
  - PlaceLocator - match place names to a specified GeoDataset
  - Standardizer - standardize address strings, street abbreviations, etc.
- Geometric & Utility
- MapObjects Helper Components

Matching an Address

- Can match single address
- Can match table of addresses

Geocode.vbp



Map Objects by Type

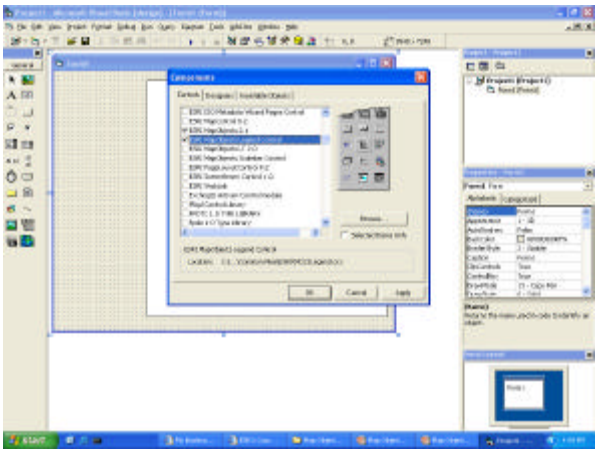
- Data Access
- Projection
- Map Display
- Address Matching
- Geometric & Utility
  - Point- single point in space
  - Points Collection- a collection of Point objects
  - Line- open geometric shape with >= 2 vertices
  - Polygon- closed geometric shape with >= 3 vertices
    - multi-part Polygon - may consist of one or more discontinuous Parts
  - Parts Collection- holds the set of Points objects that make up the parts of a Polygon or Line object
    - Most Polygons or Lines are single part shapes, and have a Parts collection containing a single Points object
  - Rectangle, Ellipse- used in Zoom In, Map resizing, etc.
  - Strings Collection- a standard collection that includes a set of unique string data types.
- MapObjects Helper Components

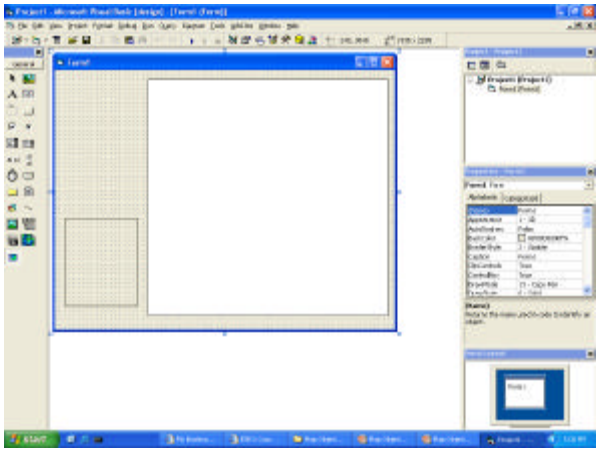
Map Objects by Type

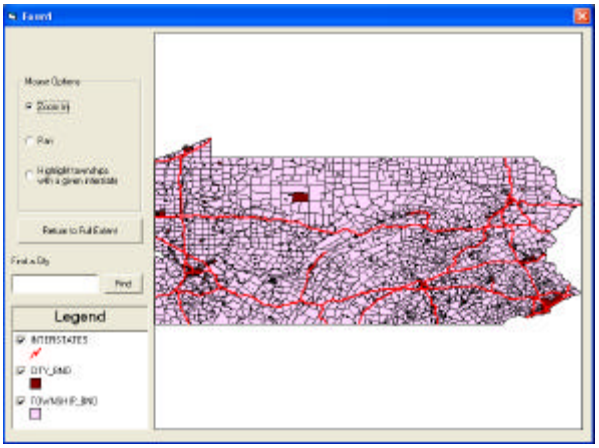
- Data Access
- Projection
- Map Display
- Address Matching
- Geometric & Utility
- MapObjects Helper Components
  - MapObjects Legend Control
  - MapObjects Scalebar Control

Creating a Map Legend

- Add Legend control to VB
- Set its map source








```
Private Sub Form_Load()

  legend1.setMapSource Map1
  legend1.LoadLegend True

End Sub
```

Acknowledgements

- ESRI
- Bob Regan
- Ken Sochats