

VBA: Generate Milepost Locations From PolylineM Routes

Contributed by Bert Granberg
17, Sep. 2009
Last Updated 21, Sep. 2009

This VBA script is designed to derive the location for mileposts routes in a polylineM feature class that has already been created.

In addition to an x and y coordinate, a polylineM routes contain a measure (m) coordinate with the milepost coordinate for every vertex. In some cases, information exists to calibrate the route at endpoints but the locations for mileposts are unknown.

The code works for single and multipart route features. It iterates through the routes and writes a record to an output .csv file for each milepost expected along the parts that make up a route. If a route starts at 0 goes to 3.777 then records will be created for 0,1,2 and 3 mileposts. X and Y coordinates are also written for each route-milepost record.

The next step is easy. Add the resulting .csv file to ArcMap, right click on it and select the 'Display XY Data' context menu item.

A set of milepost approximate locations has been created for Utah and they are now loaded into an SGID layer named SGID93.TRANSPORTATION.UDOTMilepost_Approx.

'Updated 9/21/09

```
Public Sub deriveApproxMPListFromRoutes()
```

```
'set output location
Dim outFileLocation As String
outFileLocation = "C:\routeMilepostXYnew2.csv"

'Get reference to current ArcMap session
Dim pMxDoc As IMxDocument
Dim pMap As IMap
Set pMxDoc = ThisDocument

Set pMap = pMxDoc.FocusMap

'Get reference to route layer
Dim pFC As IFeatureClass
Dim pFL As IFeatureLayer
Set pFL = pMap.Layer(1) 'Route PolylineM Layer position in ArcMap TOC
Set pFC = pFL.FeatureClass

Dim pFCursor As IFeatureCursor
Dim pFeature As IFeature

Set pFCursor = pFC.Search(Nothing, True)
Set pFeature = pFCursor.NextFeature

Open outFileLocation For Output As #1
On Error GoTo errorHandler

Dim pPolyline As IPolyline
Dim pGC As IGeometryCollection
Dim routeName
Dim pIMSeg As IMSegmentation
Dim pOutMP As IMultipoint
Dim pOutPC As IPointCollection
```

```
Dim pOutPt As IPoint
Dim pPolylinePart As ICurve
Dim pPolylinePartG As IGeometry
Dim startMP, endMP As Long
Dim p, m As Long

Print #1, "Route,MP,XCOORD,YCOORD"

Do Until pFeature Is Nothing

    Set pPolyline = pFeature.Shape
    Set pGC = pPolyline
    Set pIMSeg = pPolyline
    routeName = pFeature.value(pFC.FindField("LABEL"))

    Debug.Print routeName
    If routeName = "0210P" Then
        Debug.Print "here"
    End If
    For p = 0 To pGC.GeometryCount - 1

        Set pPolylinePartG = pGC.Geometry(p)

        Set pPolylinePart = pPolylinePartG

        If Fix(pPolylinePart.FromPoint.m) = pPolylinePart.FromPoint.m Then
            startMP = pPolylinePart.FromPoint.m
        Else
            startMP = Fix(pPolylinePart.FromPoint.m) + 1
        End If
        If CLng(pPolylinePart.ToPoint.m) = pPolylinePart.ToPoint.m Then
            endMP = pPolylinePart.ToPoint.m
        Else
            endMP = Fix(pPolylinePart.ToPoint.m)
        End If

        For m = startMP To endMP

            Set pOutMP = pIMSeg.GetPointsAtM(m, 0)
            Set pOutPC = pOutMP
            Set pOutPt = pOutPC.Point(0)
            If Not pOutMP Is Nothing Then
                Print #1, routeName & "," & m & "," & pOutPt.X & "," & pOutPt.Y
            End If

        Next m

    Next p

    Set pFeature = pFCursor.NextFeature

Loop

Close #1
Exit Sub

errorhandler:
Close #1

End Sub
```