We don’t use any shape files in our application. All data comes from SQL Server. Attached is the DataTable downloaded from SQL Server and put in an Excel Sheet for you to review.

Below is a copy of the code used to build the layer and a picture of the nScaleValue[] array. I hope this gives you enough information to reply.

I’ve also included a picture of the map (when I uncomment out the code below) so you can see I’m getting valid polygon definitions.

Thank you!!

Bob

 private void BuildDetailLayer(DataTable table, mwLegend legend)

 {

 try

 {

 MapControl.Overlays.Remove("DetailLayer");

 } catch {};

 //Setup Detail Layer

 mwMapLayer DetailLayer = new mwMapLayer(table, "name", "location", "",true); //SEE BELOW

 if (DetailLayer.ifFeaturesInitialized)

 {

 DetailLayer.ZoomLevelSet.ZoomLevel01.ApplyUntilZoomLevel = ApplyUntilZoomLevel.Level20;

 ClassBreakStyle Breaks = new ClassBreakStyle("TotalSelected");

 for (int j = 0; j < 5; j++)

 {

 AreaStyle aStyle = new AreaStyle(new GeoPen(GeoColor.StandardColors.Black), new GeoSolidBrush(GeoColor.FromArgb(100, legend.ScaleColors[j])));

 Breaks.ClassBreaks.Add(new ClassBreak(legend.nScaleValue[j], aStyle));

 }

 DetailLayer.ZoomLevelSet.ZoomLevel01.CustomStyles.Add(Breaks);

//NOTE: When I uncomment out the code below, the polygons display---but of course only one color.

 //AreaStyle aStyle = new AreaStyle(new GeoPen(GeoColor.StandardColors.Black), new GeoSolidBrush(GeoColor.FromArgb(100, legend.ScaleColors[3])));

 //DetailLayer.ZoomLevelSet.ZoomLevel01.CustomStyles.Add(aStyle);

 LayerOverlay Layer = new LayerOverlay();

 Layer.Layers.Add("DetailLayer1", DetailLayer);

 MapControl.Overlays.Add("DetailLayer", Layer);

 }

 }

 public mwMapLayer(DataTable maptable, string idcolumnname, string longitude, string latitude, bool ifpolygon)

 {

 IdColumnName = idcolumnname;

 Longitude = longitude;

 Latitude = latitude;

 ifPolygon = ifpolygon;

 if (ifPolygon)

 Polygon = longitude;

 MapTable = maptable;

 if (!SetupFeatures())

 ifFeaturesInitialized = false;

 }

 public bool SetupFeatures()

 {

 try

 {

 Feature feature;

 foreach (DataRow row in MapTable.Rows)

 {

 if (ifPolygon)

 {

 string wkt = row[Polygon].ToString().Trim();

 if (bc.Left(wkt, 5).Trim() == "MULTI")

 {

 MultipolygonShape multi = new MultipolygonShape(wkt);

 multi.Id = row[IdColumnName].ToString();

 feature = multi.GetFeature();

 }

 else

 {

 PolygonShape polyShape = new PolygonShape(wkt);

 polyShape.Id = row[IdColumnName].ToString();

 feature = polyShape.GetFeature();

 }

 }

 else

 {

 PointShape pointShape = new PointShape(Convert.ToDouble(row[Longitude]), Convert.ToDouble(row[Latitude]));

 pointShape.Id = row[IdColumnName].ToString();

 feature = pointShape.GetFeature();

 }

 foreach (DataColumn Col in MapTable.Columns)

 {

 string Value = row[Col.ColumnName].ToString();

 feature.ColumnValues.Add(Col.ColumnName, Value);

 }

 InternalFeatures.Add(feature);

 }

 }

 catch(SystemException ex)

 {

 bc.OKMessage(ex.Message);

 return false;

 }

 return true;

 }

 }



