## 报表中的地图

**ComponentOne Reports for WinForms**通过其可扩展的自定义字段的架构设计支持地图字段。Map自定义字段就是这么一个自定义字段，它使用了两个来自于**ComponentOne Studio for WPF**产品的程序集：C1.WPF以及C1.WPF.Maps，这些程序集已随着报表设计器程序一起安装。在下文中，您将了解如何通过**C1ReportDesigner**程序自定义地图字段。

注意，Map自定义字段使用到了来自于**ComponentOne Studio for WPF**产品包的两个程序集：C1.WPF以及C1.WPF.Maps。请确保在开始之前这些程序集是可用的，同时您的工程已经添加了对这两个程序集的引用。

完成下列步骤，以便在**C1ReportDesigner**程序中使用Map自定义字段：

1.运行**C1ReportDesigner**应用程序。具体请参见“从Visual Studio中访问C1ReportDesigner”章节。

1.确定Map的图标已经显示在**C1ReportDesigner**的工具栏上。如果尚未包含此图标，您需要添加下面的内容到**C1ReportDesigner.EXE.settings**文件的<customfields>部分：

<item value="C1.C1Report.CustomFields.4;C1.C1Report.CustomFields.Map" />

2.创建一个新的报表或者打开一个现有的报表。请参见“创建一个基本报表定义”的示例。

3.单击Map图标并拖拽其至报表以添加一个Map字段。

这样就OK了！Map字段主要包括：

* 地图图块和数据层
* 图例
* 样式
* 表达式自动缩放/居中以及数据跟踪
* 更多信息，请参见以下章节。下章将介绍Map自定义字段的一些重要属性。

### 图层

地图的主要部分是一个提供表示地球表面或者其一部分的光栅图像的地图图块图层，0或多个层代表着空间数据。

地图图块图层由**TileSource**属性指定。可以设置为VirtualEarth的地图图块来源（道路、航空或者二者混合显示）。地图图块层可以设置为“none”，表示地图上不显示地图图块层。这可能在当其他层，比如说KML，已经为地图展示提供了足够的数据时比较有用。

注意，除非设置地图图块层为“none”，否则当报表运行时地图图块将从网络位置加载，这将大大的影响处理速度。

除了地图图块层外，其他的图层包含在**Layers**集合中。目前我们提供了三种不同的图层类型：

* 标点层。一个标点层允许在地图上以点的方式显示空间坐标数据。标点层将为每一个数据行绘制一个标记。
* 画线层。画线层允许在表示每一个数据行的点之间绘制一条直线。
* KML层。KML（Keyhole标记语言）是一种基于XML的语言，用于描述不同的地理信息。关于KML的更多信息，请参见[http://en.wikipedia.org/wiki/Keyhole\_Markup\_Language。KML](http://en.wikipedia.org/wiki/Keyhole_Markup_Language%E3%80%82KML)图层允许向地图加载来自于本地或者基于网络的KML文件。

**指定图层的数据源**

可以为**Layers**集合中的每一个图层指定RecordSource属性（一个SQL表达式）。如果将其省略，则图层（除了KML图层外）将会从其父报表获取数据（按照当前的分组范围进行过滤）。如果指定了该属性，则会基于父报表的连接字符串使用该**RecordSource**属性。

### 目标跟踪

由Map字段显示的地图可以自动地按照显示在地图上的数据将地图居中并缩放。该行为由以下两个因素决定：

* 为整个Map字段指定的**AutoCenter**以及**AutoZoom**属性的值，以及其他相关的属性将微调自动居中和缩放（**AutoZoomPadLon**，**AutoZoomPadLat**，**MaxAutoZoom**，**RoundAutoZoom**）。
* 空间坐标数据由图层呈现，表示该数据被“追踪”。目标跟踪（比如是否一个特定的空间坐标数据片段应当用做自动居中以及缩放）由图层的**Track**属性来决定。此外，对于KML图层，也可以指定一个表达式，表示是否追踪某一个特定的KML元素。

### 样式

地图元素的可视属性大部分由样式定义。有以下几种不同的样式类型（点标记样式、线形样式以及KML项目样式）；可用的样式由上下文决定，比方说标点图层将使用点标记样式，画线层使用线形样式等等。通常一个样式可以被指定为一个数据驱动的表达式（因此实际的样式将由运行时数据决定），同时也存在一个默认的备用样式。下面将具体说说如何指定样式表达式以及表达式如何求值。

Map自定义字段包含三个样式集合：

* MarkerStyles
* LineStyles
* KmlItemStyles

这些样式可用于该Map上定义的其他图层，同样也可以用于当前报表的其他Map字段。每一个集合中的样式不仅可以按照下标索引获取，更好的办法是通过名称查找（通过**Name**属性）。当一个样式表达式计算出一个字符串，该字符串将被用做搜索匹配的样式。首先将在当前的地图中搜索，如果搜索失败，则将在当前报表的其他地图字段中进行搜索（仅匹配同类的样式；比方说，对于点标记样式，仅搜索各个**MarkerStyles**集合，其他样式也一样）。

### 空间坐标位置

标点层和画线层为数据指定空间坐标位置提供了两种不同的方式：

* 做为一个可以在运行时计算为一个经度/纬度值对的表达式对。通常将直接引用存储在数据源中相应的字段（经度和纬度）。
* 作为一个**MapLocation**指定，MapLocation指的是一个表达式（或者一个表达式的列表），可以计算出一个字符串，该字符串可以通过外部在线服务（谷歌地图）获取相应的空间坐标位置。如果指定的**MapLocation**中包含分号，则它将被处理为一个用分号分隔的表达式列表，每一个表达式将分别求值然后结合在一起用做查询。一个典型的**MapLocation**将类似于如下格式：

"Address;City;PostalCode;Country"

这将从数据源获取*Address*，*City*，*PostalCode*，以及*Country*字段，并结合在一起通过外部服务进行查询。

请注意由于网络访问速度的影响，使用MapLocation可能会是一个比较耗时的操作。因此在缺省情况下获取到的空间坐标数据将保存在一个本地的磁盘文件中。该文件的路径由**Map.GeoCachePath**属性指定。默认情况下，该文件名为“geocache.xml”，位于报表定义文件同样的目录下。不建议禁用地理信息缓存功能。

### 标点层

标点层用来显示点位置的标记，数据源每一条记录将显示为一个标记。像上一章节提到的那样，标记的位置可以由一个经纬度值或者由一个MapLocation指定。下面几点是标点层的重点。

**数据访问：**在运行时处理标点层时，报表的数据源（图层自己的**RecordSource**，或者在未指定该属性时，由当前的分组筛选的报表记录源）将被挨个遍历，将为每一个数据记录绘制一个标记。

**视觉样式**：点标记的外观由应用的标记样式决定。标点层提供一个默认的**MarkerStyle**，用做指定标记的形状、颜色等等。此外，可以为**MarkerStyleExpr**属性指定一个表达式，这样的话，在运行时将使用每一个数据记录计算这个表达式，如果计算的结果匹配当前地图的**MarkerStyles**集合中的标记样式，如果查找不到，也会尝试匹配当前报表中的其他地图，匹配到的样式将替代默认样式。（如同前面所提到的那样，样式表达式将计算出一个用做在样式集合中匹配的字符串。）

**聚类：**当几个点标记的位置相互靠近彼此的时候，它们可以被“聚”在一起成为一个单一的标记。该标记始终显示其表示的聚类在一起的点标记的数量。聚类标记的视觉样式和普通点标记有所不同，并且可能会根据其表示的点的多少有所区别。聚类样式由标点层的**ClusterStyles**集合指定，如果提供了多个样式，则特定的样式由聚合点的尺寸决定。相关的标点层属性有：**ClusterDistance**，**ClusterDistribution**以及**ClusterStyles**。

**追踪**：如果**Track**属性设置为**True**，则自动居中和缩放将包含图层中全部的点。

### 画线层Lines Layer

画线层用于在地图上的点之间绘制直线，一条直线连接表示数据记录的两个点。每一个点的空间坐标位置和标点层指定的方式一样：要么使用两个经纬度值对（每一个经纬度值表示直线的一个端点），要么使用两个MapLocation用于从网络服务获取坐标位置。以下几点描述了画线层的几个重要概念：

**数据访问**：和标点层一样，画线层允许指定其自己的**RecordSource**，或者使用经过当前分组筛选的报表数据记录。

**视觉样式**：关于样式的处理过程大体和标点层保持一致，唯一不同的是，这里将使用**LineStyles**集合而不是**MarkerStyles**集合。

**追踪**：如果**Track**属性设置为**True**，则自动居中和缩放将包含图层中全部的线。

### KML 图层

KML层在地图上呈现一个KML（Keyhole标记语言）或者一个KMZ（压缩的KML）文件。所呈现文件的文件名由图层的**KmlFileName**属性指定。该文件可以加载自一个URL、一个本地的磁盘文件或者嵌入在报表中的一个资源。如果文件不是内嵌在报表中（**EmbedKmlFile**属性设置为False），同时没有指定目录，则文件从包含报表定义文件相同的目录位置加载。

KML项目表达式：当呈现一个KML图层时，KML文件所描述的项目将按照顺序进行处理。当每一个项目加载时，图层指定的一些表达式可以进行计算并允许对这一过程进行控制，比方说基于各种不同的条件，仅加载某些特定的项目或者控制项目是否可见。此外，如果为KML图层指定了**RecordSource**，则在每一个KML项目计算其表达式之前会对这些数据进行筛选。以下是对KML项目表达式进行计算所包含的属性的详细解释。注意，所有这些表达式中，可以使用一个特殊的变量叫做**kmlItemName**，它指的是当前正在处理的KML项目的名称。

* **ItemFilterExpr：**当（且仅当）KML图层指定了一个**RecordSource**，该过滤器将在计算其他表达式之前对取数据进行筛选。比如说，如果图层的记录源包含一个Country字段，同时KML文件包含country项目，以下过滤器：

kmlItemName=Country

将确保每个KML项目，其他项目的表达式将对应于当前项目的国家计算数据。

* **ItemTrackExpr：**如果指定了该属性，将决定是否一个项目用做在地图上自动居中/缩放。如果留空，则假定为True。
* **ItemVisibleExpr：**如果指定，将用来决定一个项目是否可见。如果留空，则假定为True。
* **ItemStyleExpr：**如果该表达式计算出一个位于KmlItemStyles集合中的一个合法的样式名称（位于当前地图或者报表中其他的地图），该样式将应用到当前项目。比如说要给不同的省份按照该省份的订单总额填充不同的颜色，则可以使用该属性。
* **ItemStyle.ItemNameExpr:**KML项目样式本身包含一个计算的属性，项目的名称。这将可以控制在地图上不显示此名称，或者使用报表数据（订单总额）替代此名称进行显示。

### Legends图例

一个地图可以具有若干相关联的图例，并在地图的区域范围内进行绘制。为了方便于将图例放置在地图的区域之外，图例可以关联到报表中的任意地图字段，因此您可以添加一个仅包含一个图例的地图，用于描述另一个地图的信息。

图例包含在**Map**字段的**Legends**集合中。向该集合中添加一个项目可以添加一个图例。图例在地图区域的位置由**LegendAlignment**属性决定。Orientation属性决定图例中的项目垂直（默认值）或者水平放置。其他的几个属性可用来调整图例的外观。

图例中的项目由Items集合表示。如果图例的**Automatic**属性设置为**True，**该集合可以从当前地图中来自于非KML图层之外的数据自动产生。在这种情况下，Items集合无法被单独编辑。否则，必须手动添加图例项。

支持以下类型的图例项：

* **LegendLayerStyleItem:**表示一个图层样式。设计器允许选择一个现有的图层或者样式做为图例项表示的目标。按照选择图层样式的不同，图例项可以表示一个标记点（标点图层/样式），一条线（画线图层/样式）或者一个颜色块（KML项目样式）。
* **LegendColorSwatchItem:**表示一个任意的颜色块样本。
* **LegendTextItem**:代表任意的文本。

### 地图使用演练

在本演练中，您将看到如何向报表添加一个地图，用来显示美国各州的订单总额，按照各州数据进行汇总。原本不带地图的报表非常简洁，仅列出每一个州的订单总额。以下是全部的数据：

订单总额州名称

16325.15 Alaska

3490.02 California

115673.39 Idaho

1947.24 Montana

52245.9 New Mexico

30393.93 Oregon

31001.65 Washington

12489.7 Wyoming

您将向该报表添加一个地图，并使用颜色范围从绿色（表示该州没有订单）到黄色和红色（取决于订单总额的具体数量）填充每一个州的区域。此外，每一个州还具有一个直径和总金额按比例相关的圆形标记，同时还有一个标签表示总金额。最后您还将添加两个小的插页地图，用来表示阿拉斯加和夏威夷。

注意，该演练将使用以下文件：

* C1NWindMaps.mdb: 添加了空间坐标以及其他相关数据的C1NWind 数据库；
* us\_states\_abbr.kmz: 一个包含了美国各州边界以及州名缩写的压缩KML文件。

完成以下步骤：

**1) 创建基础报表。**

向设计器添加一个新的报表，将**C1NWindMaps.mdb**做为数据源，并使用以下SQL查询：

SELECT Orders.ShipRegion, Orders.ShipCountry, StateNamesGeo.StateName,

Sum([Order Details].UnitPrice\*[Order Details].Quantity) AS OrderValue,

(select Longitude from StateNamesGeo where StateNamesGeo.Abbr = Orders.ShipRegion) as Longitude,

(select Latitude from StateNamesGeo where StateNamesGeo.Abbr = Orders.ShipRegion) as Latitude

FROM ((Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID)

INNER JOIN (Orders INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID)

ON Products.ProductID = [Order Details].ProductID)

INNER JOIN StateNamesGeo on Orders.ShipRegion = StateNamesGeo.Abbr

WHERE Orders.ShipCountry = "USA"

GROUP BY Orders.ShipRegion, Orders.ShipCountry, StateNamesGeo.StateName

ORDER BY Orders.ShipRegion;

使用报表向导向内容节区域添加**OrderValue**和**StateName字段**，运行报表，确保其正确的显示了以上数据。

**2)添加主地图。**

请使用以下步骤向报表页眉区域添加一个地图：

1.在报表设计器中，向下拖拽报表页眉区域底部边缘，以便留出足够的空间放置地图。

2.单击Map自定义字段的图标（地球）并拖拽至报表页眉区域。

3.设置地图的尺寸为12870 ×7620twips或者相似尺寸，按照需要对其进行排布。

**3)调整地图的属性。**

按照下面的列表设置地图的属性（仅列举了非默认值的属性）：

* **AutoCenter**: false
* **AutoZoom**: false
* **CenterLatitude**: 38
* **CenterLongitude**: -103
* **ShowScale**: false
* **TileSource**: None
* **ZoomLevel**: 3

请注意因为我们要显示美国的地图，所以将按照需要手动设置坐标轴（因为我们要在右侧留出足够的空间以显示阿拉斯加和夏威夷两个插页地图）。

**4)添加点标记样式。**

打开**MarkerStyles**集合，添加一个样式。其属性设置如下：

* **CaptionExpr**: StateName& ":" &vbCr& "$" &OrderValue
* **FillColor**: 120, 255, 128, 0
* **Name**: msTotalSales
* **SizeExpr**: sqr(OrderValue / 100)

这个样式用来在每一个州的区域绘制一个圆形的标记，其尺寸大小表示订单总额。请注意填充的颜色为半透明，以便在本示例中达到更好的效果。同时请注意标记的尺寸和订单总值的平方根成正比。该样式的名称（msTotalSales）将被用来引用该样式。

您同时也可以根据自己的喜好设置其他一些属性（字体、笔画以及文本颜色等等）。

**5) 添加KML项目样式。**

将全部的州分成六组，按照不同的订单总金额范围：

* 没有任何订单的州
* 订单总金额超过1万美元的州
* 订单总金额介于1万到3万美元之间的州
* 订单总金额介于3万到5万美元之间的州
* 订单总金额介于5万到10万美元之间的州
* 订单总金额大于10万美元的州

因此您需要为每一组创建一个KML项目样式。打开**KmlItemStyles**集合，添加六个具有不同的**FillColor**属性值以区分不同分组的州的样式。每个样式的命名大体如上面的分组列表：

* **Name**: ksNoOrders, **FillColor**: 143, 188, 139
* **Name**: ks0k10k, **FillColor**: 255, 250, 205
* **Name**: ks10k30k, **FillColor**: 255, 222, 173
* **Name**: ks30k50k, **FillColor**: 255, 160, 122
* **Name**: ks50k100k, **FillColor**: 205, 92, 92
* **Name**: ks100kup, **FillColor**: 178, 34, 34

名称很重要，我们将在KML项目表达式中按照每一个州的订单总金额的值使用它们来选择一个样式。

**6)添加KML图层。**

我们的地图最重要的部分就是这个KML图层，它将用来显示每一个州的边界并使用合适的颜色对其进行填充。打开地图的**Layers**集合编辑器并添加一个KML图层。设置其属性如下所示：

* **KmlFileName：**us\_states\_abbr.kmz（指定不带路径的文件名将在报表定义文件所在的目录下搜索该文件）；
* **RecordSource:**

SELECT Orders.ShipRegion,

Sum([Order Details].UnitPrice\*[Order Details].Quantity) AS OrderValue

FROM (Categories INNER JOIN Products ON Categories.CategoryID = Products.CategoryID)

INNER JOIN (Orders INNER JOIN [Order Details] ON Orders.OrderID = [Order Details].OrderID)

ON Products.ProductID = [Order Details].ProductID

WHERE Orders.ShipCountry = "USA"

GROUP BY Orders.ShipRegion

ORDER BY Orders.ShipRegion;

* **ItemFilterExpr**: kmlItemName=ShipRegion（确保KML项目表达式，尤其是用来绘制KML项目的样式，市通过当前的KML项目相关的州的数据进行计算的）；
* **KmlVisibleExpr:** kmlItemName<>"AK"（确保主地图不显示阿拉斯加）；
* **ItemStyleExpr**:

iif (OrderValue> 100000, "ks100kup",

iif (OrderValue> 50000, "ks50k100k",

iif (OrderValue> 30000, "ks30k50k",

iif (OrderValue> 10000, "ks10k30k",

iif (OrderValue> 0, "ks0k10k",

"ksNoOrders"

)))))

以上表达式按照来自于RecordSource的每一条记录，按照不同的州计算出的OrderValue的值的合计，计算出相应的KML样式的名称（参见前一段中的样式名称定义）。之前指定的**ItemFilterExpr**属性的值确保某个州的数据已经从KML文件加载。

**7) 添加圆形标记。**

为了使地图数据更加直观，您也可以在标点层为每个州添加一些圆形标记，标记的大小和州的订单总金额成正比。您可以通过以下步骤做到这一点，首先向**Layers**集合添加一个**PointsLayer**，并移动到KML层的上方，使得在地图呈现时，该图层显示在KML图层的上方。设置该图层的属性如下所示：

* **MarkerStyleExpr:** "msTotalSales"（这将使用已经添加到地图的MarkerStyles集合的样式；按照比例确定尺寸的逻辑已经内建在该样式中，在这里我们不需要做其他额外的工作）
* **Latitude:** Latitude（使用记录源提供的空间坐标数据）；
* **Longitude:** Longitude（使用记录源提供的空间坐标数据）。

**8)添加图例：**

您需要向地图添加两个图例：一个仅仅是一个标题，位于右上角，另一个是KML项目的颜色列表，位于右下角。

1.打开**Legends**集合编辑器以添加这两个图例。

2.为了添加第一个标题，请添加一个项目，设置其**Caption**属性为“各州订单总金额”，同时保持**LegendAlignment**属性为其默认值**TopRight**。可以按照期望的效果调整其他你看到的属性。

3.为了显示颜色列表，请添加另一个图例，设置其对齐方式为**BottomRight**，之后打开**Items**集合添加以下项目：

* **Text**项目，设置文本为“颜色列表”－这将做为该图例区域的标题
* 六个**LayerStyle**项目，每一个之前添加的KML项目样式对应一个。对于每一个项目，从**LayerStyle**下拉列表选择其描述的样式－这将从选中的样式中获取相应的值，并自动地设置大部分其他属性。您唯一所需要动手设置的是项目的文本－为每一个样式设置合适的描述，从“无订单”，到“小于$10K”，再到“$10K－$30K”，等等，一直添加到“大于等于$100K”

**9) 添加插页地图。**

我们将为阿拉斯加和夏威夷添加两个额外的插页地图。按照以下步骤：

1.单击Map自定义字段图标，在主地图的左上角绘制两个小一些的地图，上面是阿拉斯加，紧挨着下面一点是夏威夷。

两个新增的地图将重用主地图定义的样式，因此我们不必添加新的样式到插页地图的样式集合。

2.图层分布和主地图一致，我们为每一个插页地图的Layers集合添加两个图层：

* 一个标点层，MarkerStyleExpr设置为“msTotalSales”，同时MarkerVisibleExpr设置为

StateName = "Alaska"

（在夏威夷的插页图中，设置为“Hawaii”），其他的属性和主地图保持一致。

* 一个KML图层，全部的属性（包括RecordSource属性）从主地图的KML图层复制过来，最后将ItemVisibleExpr属性设置为

kmlItemName="AK"

（对于夏威夷，设置kmlItemName="HI"）。特别需要提到的是，即使**ItemStyleExpr**计算出的样式名称引用自其他地图的样式集合，该样式依然能够正常工作。

3.最后，为每一个插页地图添加一个图例，分别显示这两个州的名称。

大功告成。现在可以运行报表，确保我们以上的设置能够正常工作。