第一章了解 MapObjects

MapObjects 是一套制图软件集，它使程序员能够把地图加到应用程序中去。

通过 MapObjects 你可灵活地建立适合用户的地图接口。在小内存空间中，你能用多种工业标准程序环境之一去建立应用程序，你能够联合使用 MapObjects 与其它软件去实现地图与用户信息的联系。

1. 1元素软件

大部分商业软件在最近的版本中极度壮大，诸如字处理及报表等程序，其最初仅有几兆，目前却严重消耗磁盘空间。这就要看最新的计算机能否快速执行最新版本软件以至于不落后于旧机器使用老软件的效率。

由于操作系统变得越来越复杂及用户对更多功能的要求也日趋强烈，程序也随之壮大了。然而典型用户也仅充分使用了这些大软件的 10%-20%。那些没有用到的功能象一个大包袱，严重消耗了系统资源和磁盘空间。

如何打破这种恶性循环?元素软件(**Component software**)是一技术上的重大突破。它提供了一种解决办法。

元素软件的原理是把大的桌面应用软件的功能打碎成部件。开发人员可跟据需要恰当地将一些部件组合成工具箱，用它建立专用软件。这种部件叫目标(**objects)**。所用平台就是可视化程序语言。其结果就是精炼地程序快速经济地运行并且对特定的市场有很强的适应性。

1. 2MapObjects 的功能

通过 MapObjects 你可完成以下甚至更多功能：

- 显示一张多图层地图(道路，河流，边界)
- 放大，缩小，漫游。
- 生成图形元素，如点，线，圆，多边形。
- 说明注记
- 识别地图上被选中的元素。
- 通过线，方框，区域，多边形，圆来拾取物体。
- 拾取距某参照物特定范围内的物体。
- 通过 SQL 描述来选择物体。
- 对选取物体进行基本统计。
- 对所选地图元素的属性进行更新，查询。
- 绘制专题图。
- 标注地图元素。
- 从航片或卫星图片上截取图像。
- 动态显示实时或系列时间组数据。
- 在图上标注地址或定位。

本书开发了一系列通过 VB，OLE，WinAPI 扩展 MapObjects 的办法。

MapObjects 可执行许多基础制图功能。但它不能执行某些高级功能。如，高质量地图输出，地图坐标系投影，表面模型或网络分析等高级空间分析。以及拓扑编辑。你可利用 ESRI 的其它产品如 ARC/INFO，ArcView 来实现高级功能。

1. 3MapObjects 简介

MapObjects 包括一个 OLE 控件(OCX)叫做地图控件(**Map control**)和一组(三十多个)OLE 目标(objects)。它适用于工业标准程序环境。如 VB4.0，Delphi2.0,PowerBuilder,MS Access 等。

MapObjects 不适用于最终用户。它是为程序开发者设计的。程序开发者可利用 MapObjects 开发应用程序并把这些程序提供给下一级用户使用。

MapObjects 运行于 Win95 或 Win NT3.51 或更高。

最近，Mierosoft 开始把 **OLE controls** 称作 **ActiveX controls**。本书将元素称为控件(OLE controls 或 OCXs)，但这些术语可与新术语 ActiveX 互换。

MapObjects 和 OLE containers

OLE Automation object 是一种可编程目标，它可通过 OLE Automation server 来通讯。OLE custom control 是一种软件元件，它可通过 OLE container 实现特定功能。OLE container 和 OLE Avtomation Server 就是 VB，Delphi，Access，VFro

OLE custom control 有事件，属性和方法。

OLE Automation objects 有属性和方法。

在 OLE Container 中(如 VB)，你能联合使用 Map Objects 的目标和其它 Custom controls，及利用别的程序中的 OLE Automation objects(如 office)去建立应用程序。

部件软件的优势

- Map Objects(通常叫 OCXs)可容易地用 OCXs 建立程序而免去 C++ 冗长的代码，你可发挥专长于设计，工程，制图，而不是软件。
- 你的 Map Objects 应用软件不多占用内存空间。
- 比其它 Win 支持的制图软件速度更快，MapObjects 直接建立在 MFC 库上且优化执行。
- 你有极大的灵活性去建立用户接口。你可在应用程序上建立几乎所有在 Win95 上见过的接口。

MapObjects 的另一特点是，地图可以是程序中的主要元素也可是附属品。因此 MapObjects 尤其适合 vertical 应用软件。例如， 你的软件可能主要是商用数据库，当查询时可在表单上高亮度显示与查询内容相关的地图。

1. 4使用 MapObjects

把一张地图加到 VB 的表单上十分容易，如图

****

****

用 MapObjects 编写程序的步骤:

- 从 CD-ROM 装入 MapObjects
- 启动 VB 在"工具"菜单下打开 Custom controls 对话框。如果安装成功你将看到 ESRI MapObjects 的全称。用鼠标点 check box (选 MapObjects 可用)。你会看到 MapObjects 图标被加到 VB 工具栏内。选地图控件，把鼠标移到表单中，拖动十字丝把矩形框尺寸调到适当大小，释放鼠标，你的表单上就会有一个地图控件。
- 通过地图控件属性框或通过对 Data connection 和图层目标的编程，你可向地图控件中增加地图数据。
- 你还可以继续添加其它控件，如命令按钮和工具条，或编写代码来调用属性，事件和方法。
- 编写，调试，编译。

安装 MapObjects

一旦从 CD-ROM 中成功装入，下一步就是向 VB 工程中增加 MapObjects
****
****

当你打开一个 VB 的工程，工具档中包含了一部份你已购买的 VB 控件。但并不是全部。你会发现制定缺省控件装入从而把 MapObjects 包括进去是十分方便的。

为达到上述目的，你可向 C:\Program Files\Microsoft Visral Basic\Avto 32 ld.vbp 中加入所需控件。一但你把 MapObjects 加入，在每次开始新工程时，地图控件将被自动装入。

你可自己制定一次装入控件的数量。按需要决定控件的数量是一个好习惯。这样会使你的工程装入的快点且节省内存。

加入一地图控件

你可向任意 VB 表单中加入一个或多个地图控件。
****
****

当你向表单中加入一地图控件，最初它就象 VB 的图片框控件， 然而它的资源并不是位图而是矢量或栅格数据的图层。

加入一图层

你可通过地图控件的属性窗或通过写代码来加入图层。
****
****

你可通过地图控件属性窗口增加图层也可采用在图层属性窗口中对图层目标赋属性的办法加如图层。

地图控件属性窗和图层属性窗是连接地图控件和图层目标的某些属性的通道。

在实践中你可通过写代码来增加图层，因为你通常不能预言你的下级用户新需接的图层在哪里。

下面是加入图层的例子：

```
Dim dCON As New MapObjects.DataConnection
Dim curLayer As New MapObjects.MapLayer
dCON.Database = "c:\Data"
curLayer.GeoDataset = dCON.FindGeoDataset("Counties")
map1.Layers.Add curLayer
```

这是一段通用程序，下一章我们将进一步讨论 MapObjects 的数据通道(data access)目标，将介绍更多的增加图层的程序。

编写代码和增加属性

以下是如何使用 VB 代码编辑器和设置属性：

****
****

象其它所有控件一样，当在地图控件上打开 VB 代码窗时，你就会看到关于地图控件的所有事件的代码结构。

在最后一部分，你会看到在地图控件属性表中可快捷设置地图控件的属性。

注意，在标准属性窗口中，地图控件的一些属性是不可见的。这些属性是只读的。我们在后面会涉及。

取得 MapObjects 的帮助

MapObjects 具有在线帮助系统，它包括 MapObjects 每一目标的属性，方法和事件，以及常量的详细说明。

大多数帮助的主题下都有一小段程序例子，它可在 VB 下执行。

下面有三种办法可得到帮助：

- 你可点工具栏中地图控件并按 F1
- 可通过 VB 的目标浏览器与 MapObjects 的帮助建立联系。
- 可把 VB 编辑窗口的文本游标移到 MapObjects 的某一目标，属性或事件上按 F1

一种最好的学习 MapObjects 的方法就是学习例子。以下是使用例子的步骤：

- 在帮助的主题下点例子。
- 打开一新 VB 工程并加上一地图控件。
- 当表单是活动状态时按 F7 打开代码编辑器。
- 读例子的结构，并向表单中加入特定控件。
- 从例子中粘贴代码到代码编辑器中。
- 按 F5 或选择 RUN。你的例子将被运行。

运行工程

你可编写一小段应用程序。加入地图控制并加入一个或多个图层。在编写时你不会看到任何图层显示出来。但当程序运行时，你就会看到图层将掌满显示区域，如果没制定符号属性(颜色、大小 、型)，则缺省设置将自动启用。

1. 5使用 OLE Automation objects

除了地图控件以外，MapObjects 包括三十多个 OLE Automation objects，这些目标是 MapObjects 区别于同类产品的制图软件部件，它们提供了灵活性和多功能。
OLE Automation objects 是编写交互的高效应用软件的关键。

使用目标浏览器

OLE Automation objects 在打开 VB 之初并没有呈现出来。但可在目标浏览中看到它们，点 View 菜单中 object Browser，在库/工程下拉框中选"Mapobjects-ESRI Mapobjects"你就会在类/组件列表框中看到 MapObjects 的 constants 和目标。

- 点任意目标，在方法/属性列表框中就会有增加。
- 点任意方法或属性，你就会在目标浏览器底部看到一小段描述。
- 点 question 按钮，就会得到 on line 帮助。

引用 OLE Avtomation objects

大多情况，用 MapObjects 就要用地图控件。地图控件是一种可视化元素，然而有一些用 MapObjects 编制的应用程序并不需要地图控件。你可以在 VB 的 Reference 窗口选中 MapObjects OLE Automation objects 而不用增加地图控件。在 VB 工具
注意，如果你把 MapObjects 加到 VB 的工具栏中，也就自动地引用了 OLE Automation objects

声明目标

当你声明或建立一新目标时，最好写全名，例：

```
Dim rSet As New Mapobjects.Recordset
```

而不是

```
Dim rSet As New Recordset
```

这不仅可提高运行效率，也避免了潜在的命名矛盾。如 MapObjects 的 Recordset 目标和 VB 的 Recordset 目标。

使用常量

本书涉及了所有 MapObjects 可用的常量，在设置变量或属性时可使用常量而不是数值。常量代表的具体数值请参阅参考手册和在线帮助。

明确变量类型

如果你还没有做准备，你就应该在 VB 的工程中声明变量和目标的类型。从 Tools 单中选 Options 项，在 Options 对话框中点中 Require Variable Declaration，在任何一个新表或模块中都会具有这个设置。

在声明段，你的微不足道的变量和目标声明工作将换来少测试和高效的表。

1. 6MapObjects 的数据源

你可通过 MapObjects 使用形文件，图像文件，属性表或通过 ESRI 的专用数据库引擎连接的专用数据库。

形文件是地图数据的矢量形式，图像文件是栅格图像或尤指航空或卫量的畸变图像的纠正照片，属性表是可用 ODBC 装入的任意格式。 专用数据库是网络上通过 ESRI 专用数据库引擎连接的 UNIX 服务器。

形文件适用于中小型地图数据。而大型数据(省，国家道路网)，你就需使用专用数据库。

用 MapObjects 编写的软件是可伸缩的。最初你可用形文件。当用户需要与大型数据库连接时，几乎所有代码都可被移到与专用数据库连接后的工作中，你仅需修改打开数据源的几行代码。

形文件

形文件是 ESRI 提供的存储地理数据的矢量格式。这就意味着地图元素以 X，Y 形式出现。其坐标系是笛卡尔坐标。注意，笛卡尔坐标与屏幕坐标有所不同。

每一元素的几何形状以包括一组矢量坐标的形的形式存储。其属性存放在与形文件相连的 dBASE 的记录中。

以下是得到形文件的不同方法。

- 购卖商业地图数据。
- MapObjects 提供的光盘。
- ARcView，ARC/INFO 等 ESRI 产品的数据。
- 编写程序从其它格式中转换。

一个形文件由三种文件组成，主文件(*.shp)包含几何形状。索引文件(*.shx)包含数据的索引。数据库文件(*.dbf)包含形的属性,你可以修改字段的定义。

每一个形文件包含一种类型(点 弧 多边形)

- 点有一个(X，Y)坐标和一个属性。

- 弧段包含一条或一组(可连，可不连)的多义线。一条多义线是一组有序结点。每一弧段有一个属性记录。
- 多边形包括一个或多个边界，一个边界是一个无交叉点的闭合环，一个边界可嵌于一多边形中而形成环形。边界的方向决定它是否代表区域内的面积。每一多边形有一属性记录。

形文件通过 ODBC 读入，ODBC 在装 MapObjects 的同时被装入并注册。

ARC/INFO 用户应注意形文件中弧，多边形的定义不同于 ARC/INFO coverage 中的定义。形文件无拓扑，因此，形文件允许你集合简单元素来合成元素。如，把几条 polyoline 会成 arc。通过形文件，你可快速显示图形并具有一简单数据模型; 以简单数据模型换取快速显示，这使得对形文件进行拓扑编辑或高级分析变得十分困难。

形文件是一种通用格式，如果你想编写程序直接读写形文件，你可以从 ESRI 网点 http://www.esri.com 上下载 ArcView V 2.0 形文件描述。

图像(Image)文件

你可通过 MapObjects 编写应用程序来显示多种图像文件。在地图中图像多来自航空照片和卫星图像。

图像文件依靠带有灰度值或色标的一组像元来表示图片，这些像元无属性连接，其坐标系统与形文件不同。

你可把图像文件精确重叠于大地坐标的形文件。MapObjects (或其它 ESRI 软件)用 world files 来配准图像。

一个 world 文件是一个简单的文本文件，它包括数学参数来定义转换关系，其公式为：

x' = Ax + By + C

y' = Dx + Ey + F

x' 像元在地图上的计算坐标值 X

y' 像元在地图上的计算坐标值 Y

x 像元列数。

y 像元行数。

A X 轴上像元的尺寸。

B,D 旋转关系项。

E 负的 Y 轴上像元的尺寸。

C,F 左上角像元中心的 X Y 地图坐标。

注意 E 为"负"值，因为，形文件坐标与图像坐标 Y 方向正向反。

world 文件是包含 A，B，C，D，E，F 值的连续行文本文件。

注意 MapObjects 不支持图像旋转。这样 B，D 的值在 world 文件中是被忽略的。为果需要旋转，你可用 ESRI 的 ARC GRID

以下是 MapObjects 支持的图像文件格式。

名称	描述	扩展名	World file 扩展名
BMP	Windows bitmap	*.bmp	*.bpw
TIFF	Tag image file	*.tif	*.tfw
SUN	Sun raster file	*.sun	*.snw
ERDAS	ERDAS GIS or LAN	*.gis	*.gsw

IMPELL	IMPELL bitmap	*.rls	*.rlw
BIL	Band interleaved by line	*.bil	*.blw
BIP	Band interleaved by pixel	*.bip	*.bpw
BSQ	Band sequential	*.bsq	*.bqw

属性表

用 MapObjects 编写的应用程序，可通过一种关系与外部属性表相连。关系是连接元素表(元素表可是形文件的 dBASE 表，也可是从 SDE 层中得到的表)与属性表的表。为了这种连接，可安装 ODBC。这种关系留存于应用程序运行期间，它不会被写入文件中。

要建立这种关系。你要确认一个元素表的某一字段，一个要与之建立关系的属性表和该属性表的一个字段。属性表的相关字段必须是 primary key 或允许在其上建立一独一无二的索引。有一例外，在少于 100 个记录的小型元素表上可建立无特殊字段的关系。

一旦建立了关系，它就在元素表上建立了一种纽带，你可通过属性表的字段查询属性，但你不能在 MapObjects 中通过 SQL 表达式向里面增加数据。

空间数据引擎

如果你采用大规模地图数据组来组织工作，见意考虑使用空间数据引擎(SDE)，一种高性能制图数据服务器。

通过 SDE，空间数据可存放于 UNIX 服务器上。用户的 SDE 应用程序可基于 UNIX 或 WIN 环境被编写，SDE 提供软件开发和数据管理能力：

- 管理大规模地理数据，提供地图无缝显示。
- 通过某种商业关系数据库存储数据。
- 通过一组高效的尖端空间数据操作来查询空间数据。

SDE 包括一个 C 语言应用程序接口(API)，它提供最大能力的执行效率和极大的灵活性。下一章，我们将详细解绍与 SDE 的连接。

1.7纵览 MapObjects

MapObjects 包含一组具有属性，事件和方法的目标。

你已经看到你可通过地图控件快速建立一简单应用程序，但实际的应用中，需要使用 OLE Automation objects。当你初次使用 MapObjects 你应了解这些目标、属性和方法。这些对于 MapObjects 的全部组织是非常有用的。

MapObjects 的目标分为四组：

- 数据通道目标组
- 地图显示目标组
- 几何图形目标组
- 地址匹配目标组

数据通道目标组

通过数据通道目标组，你能建立与地图数据的联系。增加属性值，从地图元素上反馈属性信息。数据通道目标组由以下部件组成：

数据连接(Data Connection)目标是 MapObjects 通向地图数据的通道。它包括属性和方法来建立与地理数据集合(GeoDatasets)的联系。

地理数据集合(GeoDataset)目标代表制图数据并可引用图层。它可引用形文件或 SDE layer 的数据。

地理数据集合是对于一个数据连接的所有地理数据集合目标的总合。它是一特定文件夹中所有形文件或 SDE 数据库中所有 SDE 层。

记录集合(Record set)目标代表一个图层的记录。如果你做了一个选择集，它就代表所选记录。它类似于数据库指针。

TableDesc 目标给你关于与记录集相连的表的字段的信息。

表(Table)目标是一个只读数据通道目标。它代表来自 ODBC 数据源的一个表单。你可增加一表作为与图层目标的关联或为了大批地址匹配。

字段集合包括记录集目标的字段目标。

统计目标代表关于一个记录集的简单统计信息。你首先应用一方法计算关于记录集的统计值，然后可在统计目标中检查结果。

字串集合是一组标准字符串数据类型集合。你可通过它从其它目标中取值来自接口控件(如列表框)中增加数据。

地图显示目标组

通过地图显示目标组，你能用符号或专题描述画一张地图。你也可加入图像做为背景，在地图上显示动态数据。地图显示目标组由以下部件组成：

地图控件使你能显示图层、图像层，和动态跟踪图层目标，你可编写代码来控制鼠标驱动绘图事件，设置显示参数，用方法可画元素，闪烁选择的元素，计算点与元素的距离，输入线，圆，三角。

层(Layer)集合是服务于地图控件的图层目标和影像层目标的集合。

图层目标代表带有一些显示属性的地理数据集合目标。它可让你处理专题地图，此目标有几个方法来查找和选择地理元素。

图像层目标代表一作为地图控件上的背景的影像文件。

动态跟踪图层(Tracking Layer)目标让你能动态拖拽元素而无需重显。这对实时数据获取是十分理想的(如 GPS)。它也可用于显示基本几何形状(如三角、圆)和描述性文本。它们都不是地图数据的一部分。

Geo Event 目标代表可加到 Tracking Layer 目标上的点元素。

符号(symbol)目标是广泛使用的目标，它影响如何在地图上显示元素的许多方面。其属性包括：颜色、字形、大小、形状。

文本(Textsymbol) 目标代表文本的某些属性(如准线、字型)

ClassBreaksRenderer 目标使你能在图层目标中通过分类的办法依数值字段显示元素。

ValueMapRender 目标使你在图层目标中通过特殊字段中单独的值，用符号来显示元素。

LableRenderer 目标使你在图层目标中，依元素的某一字段的属性标注文本。

几何图形目标组

几何图形目标组提供几种功效：依从图层中选择的元素反馈几何信息；向图层添加几何目标；向地图中画几何目标而不更新图层。几何图形目标组由以下部件组成：

矩形(Rectangle)目标经常用来设置和反馈地图范围，也用来画矩形。

点集合存贮线和多边形目标的坐标。

点目标代表具有 X、Y 坐标的点。

线目标代表地图上的一条线。

多边形目标代表多边形。它的头一个点和最后一个点在它的点集合上是相同的。

椭圆目标代表椭圆和圆。

地址匹配目标组

地址匹配目标组让你进入一图层上的某个地址，该地址具有街道和地址范围并返回一个位置，你也可发现十字路口的位置和地名。地址匹配目标组由以下部件组成：

地址匹配(Address Matcher)目标让你列出含有地址范围的道路中心线的地理数据集合, 并具有为单个或一批地址匹配的方法。

地址目标代表关于 Address Matcher 目标的方法中标准化地址。

定位目标(Place Locator)让你列出带有地名的地理数据集并通过一个方法找出地名的位置。

地址位置(Address Location)目标包含一段代码指示是否(或如何) 一个地址被解决如果已被解决指出与地址匹配的图型位置。

1. 8 moView 应用程序

我们提供一种叫作 moView 的应用程序来展示 MapObjects 许多功能的优点并提供模板，在这本书的许多章节将通过代码片段和来自于 moView 及其它例子的应用来揭示 MapObjects

你可在 MapObjects 光盘的例子目录中找到 moView，如要学习只需考贝 moView 文件夹到你的系统目录中。

moView 是一种普通简单的制图应用程序。它的接口围绕着地图控件而调整，这种类型叫作地图中枢(map ceneric)，也许在你的应用程序中地图很象是一个附件而不是核心。

在应用程序中，没有硬编译代码(Hard coded)的数据源，你可使用任何 MapObjects 可接受的数据。moView 展示了许多事件、属性和方法。并提供一个框架使你在开发程序时可踊跃前进。

你可为某特定市场建立纵向应用程序。这时你可使用 moView 片段，它给你的应用程序做向导。并可扩展为特定应用程序所设计的接口。

运行 moView

一但在 VB 内装入 moView，你可按 F5 或点 Run 菜单中的 Start
当你打开 moView 你就会看到一个主表和一个空地图显示区，你可点工具条中第二个按钮(地图目录)来引入图层，地图目录表很象 VB 中地图控件属性表单。

moView 命令集

moView 的主要命令被装在主表的工具条中。
打印，向缺省打印机输出地图控件上的当前显示。
查找，可产生一元素查找表，通过它你可根据属性来定位元素。
空间选择(Spatial select)，将产生空间选择表，通过它可实行 MapObjects 的所有空间寻找方法。
地址匹配(Adress matching)，可产生地址匹配表，通过它可执行基础地址匹配。
撑满(Full extent)，在地图控件中撑满地图包括所有图层中的所有元素。
放大(Zoom in)，开窗放大。
缩小(Zoom out)，以鼠标点为中心降低地图比例二倍。
漫游(Pan)
标注(Edeneify)，标注鼠标点或附近。
图形(Graphics)，激活图形工具条可画几何图形。

框架开发

用 VB 建立一地图应用程序的用户接口是一项特殊的要求。因为 VB 提供的控件最适合交互式文本和表，对于图形仅为边缘插图功能象地图的图式。moView 的一个重要的中心任务是解决建立地图用户接口问题。
我们不选用第三方控件，因为我们不能预言或假设你除了 VB 提供的控件以外使用什么控件。为建立地图应用程序的用户接口。 你可选择任何一个厂商提供的控件。
对于 moView 应用程序的源代码和表你都有通道与连接，无论你有无 VB 经验，moView 的代码对你都是有价值的。

moView 的方针

你会发现 moView 应用程序在 VB 下安装了一些程序标准，这些应用程序提高了应用程序的健壮性、可读性和可靠性。你的选择是由你和应用程序的大小和特性决定的。对于一成功的应用程序开发来说。程序标准是一个基本部份。
在 moView 中使用的一些程序方法：
用控件的标准前缀的命名约定。
整个代码中大量的描述性注释。
变量范围限制和全局变量限制。
在开始阶段装入并隐藏表来优化执行。

向标准模型中压缩数据和方法。

第二章

使用地图和图层

使用 MapObjects 的起点就是向应用程序中加入地图并向地图中加入数据。

向应用程序中加入地图十分简单，只需把工具箱中的地图控件加道 VB 的表单中，象其它控件一样，你可调整它的大小和位置，并可同过 VB 的属性窗口或地图控件属性窗口来改变属性。

通过地图控件属性窗口加入形文件十分方便，当你在应用程序中使用 MapObjects 时，还会发现使用图层是十分必要的。

最初，你会对向地图中加入图层感到不适应，你不得不研究大量的 OLE Automatiion objects、SDE 层和影像文件。但渐渐你会发现 MapObjects 的数据通道和地图显示目标具有极大的灵活性和多功能性。

本章将涉及以下内容：

- 如何在地图上表示层。
- 层的次序。
- 向地图中加入形文件。
- 向地图中加入图像文件。
- 建立地图用户接口。
- 设置地图属性。
- 数据连接。
- 修改图层。
- 动态跟踪层。

本章将涉及以下目标：

GeoDatasets 集、DataConnection 目标、GeoDataset 目标、地图控件、图层集、图层目标、影像层目标、动态跟踪层目标、GeoEvent 目标。

使用 MapObjects 向地图中添加数据有三种方式：

- 通过建立 DataConnection、GeoDataset 和图层目标及向地图目标的层集中添加数据的方法加入矢量地图数据。
- 通过建立影像层目标及向地图目标的层集中影像层的方法显示影像地图数据以作背景。
- 通过使用动态跟踪层目标和添加 GeoEvent 目标的方法实现动态跟踪。

在地图上画层

地图包含许多层。现在我们要讨论 MapObjects 的各种层目标和如何在地图控件中管理图层。

用户眼中的图层

地图的最上方是动态跟踪层，最下方为地图控件，中间为层集。层集中图层目标和影像层目标可以任意顺序排放，但通常影像层显示在最底层作为背景。

以层方式工作十分有意，因为它十你很容易的选取同类地物。

程序员眼中的层

以下是影响显示地图数据的关键目标：

- 地图控件是显示图层的平台，它有两个重要的属性：层集和动态跟踪层目标。
- 层集包含图层目标和动态跟踪层目标。它们的顺序决定在地图控件中的相互覆盖关系。
- 图层目标代表矢量数据。
- 影响层目标代表栅格数据。
- 动态跟踪层目标显示实时数据。

地图控件的关键属性是层和跟踪层。当你向表单中假如一地图控件，你也同时建立了一个空层和空跟踪层。

另一重要属性是 Extent。它决定地图的显示范围。

用 Refresh 方法在地图控件上画层。当你执行下列操作时 Refresh 方法将自动执行：

- 向层集中加入图层或影像层。
- 使用 Pan 和 CenterAt 方法。
- 更新地图控件的 Extent 属性。
- 使用 Clear 或 Remove 方法。

注意，有些操作不会引发 Refresh 方法，在编写程序时应在这些操作后加上 Refresh 方法一使操作更新显示。

标准控件的属性和方法

如果你是 VB 的用户，你应了解一些标准属性和方法：Container、DragIcon、Index、Left、Height、Drag、Setfocus 等，这里我们还将介绍一些 Mapobjects 的专门属性。

层集

层集是地图控件的重要属性，包含图层和影像层。

使用 Add 方法可向层集中加入图层或影像层，无论增加了图层还是影像层，FullExtent 属性都会使地图撑满。

如果你想使全图重新显示，可设置 Extent 属性为 FullExtent

```
Set map1.Extent = map1.FullExtent
```

Clear 和 Remove 方法可删除层集中某层。例如：

```
Map1.Layers.Remove (4)
```

MoveTo、MoveToBottom 和 MoveToTop 方法可改变层集中侧的顺序。

层的表示可以有层名和索引两种方式。例如，下面的例子中索引号为 2，层名名叫 Hydrography

```
Map1.Layers.Item(2).Symbol.Color = vbBlue
Map1.Layers(2).Symbol.Color = vbBlue
Map1.Layers.Item("Hydrography").Symbol.Color = vbBlue
Map1.Layers("Hydrography").Symbol.Color = vbBlue
```

图层目标

图层目标代表矢量数据。你可以下面的方式建立新层：

```
Dim mLayer As New MapObjects    MapLayer
```

当你通过 DataConnection 目标的 FindGeoDataset 方法把图层加到 GeoDataset 目标中时，以下操作将自动执行：

- Extent 属性被更新为地图的最大范围。
- Records 属性被分配了一 Recordset 目标。
- ShapeType 属性将依据形文件类型被设成 moPoint、moLine、moPolygon
- Symbol 设为缺省，并随即赋予颜色。
- 如果 FindGeoDataset 方法成功，Valid 属性将被设成 True

- Visible 属性设成 True

Renderer 属性将设为 Nothing,等待你设定其中的一个 Rerderer 目标：ClassBreaksRenderer、ValueMapRenderer、DotDensityRenderer 或 LableRenderer
Maplayer 给你提供了一些强有力的方法以改变 Maplayer 的属性数据。这些将在第五章里讲述。
影像层目标
影像层目标表示你的层集中的一个图象文件。象图层目标一样，可以如此创立一个新影像层目标：

```
Dim iLayer As New Mapobjects ImageLayer
```

然后，你可在层集里用 Add 方法往你的地图画面里加入图象文件。你可能还要使用层集里的 MoveToBottom 方法，这样图象就不会挡住其它的层。
当你将影像层附加图象文件时，Extent 属性就会被更新，最大限度地反映地理范围。在同一地理坐标上，如果你想使用其它图层，图象文件应有一个相关的配准文件。
反馈图层类型
在层集里用 Add 方法加入图层或影像层，如方法成功,就会反回 True
一旦层集里包括了一些层目标，你必须随时掌握各种层的类型。下面是在层集中返回层类型的例子：

```
Dim Layer As objects
For Each Layer In Map1.Layers
If Layer.LayerType = moMapLayer Then
MsgBook "layer" & Layer.Name & "is a map layer"
ElseIf Layer.LayerType = moImageLayer Then
MsgBook "layer" & Layer.Name & "is a image layer"
End If
Next Layer
```

注意，我们必须将层声明为目标，这样 Visual Basic 可在运行时接受此定义。
在上面例子中，还用到两个常量：moMaplayer 和 moImagelayer，这增加了程序的可读性。你可在 MapObjects 帮助屏上或在 MapObjects 程序员手册上查到所有常量的值。
全面检查图层
在上面例子的基础上，使用层集索引全面检查图层属性：

```
Dim i As Integer
For i = 0 To Map1.Layers.Count - 1
If Map1.Layer(i).LayerType = moMapLayer Then
If Map1.Layer(i).shapeType = moPoint Then
MsgBook "Layer number" & i & "is a map layer and contains points."
ElseIf Map1.Layer(i).shapeType = moLine Then
MsgBook "Layer number" & i & "is a map layer and contains lines."
ElseIf Map1.Layer(i).shapeType = moPolygon Then
MsgBook "Layer number" & i & "is a map layer and contains polygons."
End If
ElseIf Map1.Layer(i).LayerType = moImageLayer Then
MsgBook "Layer number" & i & "is a image layer."
End If
Next i
```

图层显示
图层在地图控件中的显示顺序与其在层集中索引的顺序相反。
新增图层的索引号总是 0，以有图层的索引号按顺序递增。这与 VB 的一些集合有所不同。
在地图控制单中,Layers 是按 Layers 集里索引值的例序排列来调用的.
Drawing 事件
当你在地图上调出图层后，一系列事件就会起动。你可将程序放到这些事件后面，并可让用户取消调出的大的地图层。
当你在地图控件中使用 Refresh 方法,就会出现如下情况：
1. BeforeLayerDraw 事件起动。
2. 层集按索引相反顺序调出。
3. AfterLayerDraw 事件起动。
4. BeforeTrackingLayerDraw 事件起动。
5. 如果任何 GeoEvents 加入到 TrackingLayer 中，他们就被画出。
6. AfterTrackingLayerDrow 事件启动。

这些 Drawing 事件的一个使用方法是往地图上画一些几何图形，如线、圆和在地图上说明注记，另一使用方法是调出层集或 TrackingLayer 之前来检验状态。
如果你的用户在地图上已启动了一种方法，而它又占用了太长的时间，你可按 Escape 键取消操作。你可赋予地图控件的 CancelAction 属性以下三种值
- moCancelNone 是说按 Escape 键将被忽略。
- moCancelMap 将停止所有层的调出。
- moCancelLayer 将停止当前层的调出。

按 Escape 键启动 DrawingCancelled 事件。将程序放到事件后，以便应用程序可通过恰当的方法处理绘图过程的末端。
在你的应用程序中控制地图的调出速度的方法是在地图控件中设置 RefreshCount 属性。Mapobjects 在缓冲器里调出地图，当达到一定矢量数目时更新地图显示。缺省情况下，此属性为 10,000。降低这种特性在较慢的计算机上可能会明显提高效率，增加这种特性将加快一些地图控件的调出。
利用 Win API 扩充 MapObjects 应用程序
MapObjects 提供了一种方法来使用经常出现在 Window 应用程序接口程序中的两个变量 hDC 和 hWND
hDC 是一句柄，一个唯一 Windows 分配给表面设备的 ID 号，如屏幕或打印机，通过它你的应用程序可产生可视操作。
你可通过 Windows API GetDC 函数可以得到自己的 hDC，但使用完，一定要用 ReleaseDC 功能退出。一旦你有了 device context，你可用适合你的应用程序的 Windows API 图表来使用它。例如，在一个应用程序里，这个应用程序能够在地图点位上提供图符设制，你就可以象如下程序：

```
Private Sub mapMain_BeforeLayerDraw(ByVal index As Integer, ByVal hdc As Long)
If index = 0 Then
RenderIconicMarkers hdc, mapMain.Layers(index), picLogo
End If
Private Sub RenderIconicMarkers(hdc As Long, layer As MapLayer, pic As
PictureBox)
Dim recs As MapObjects.Recordset
```

```
Set recs = layer.Records

Dim shpfield As MapObjects.Field
Set shpfield = recs.Fields("Shape")

Do While Not recs.EOF
Dim pt As MapObjects.Point
Set pt = shpfield.Value
DrawIconicMarker pt, pic, hdc
recs.MoveNext
Loop
End Sub

Private Sub DrawIconicMarker(pt As MapObjects.Point, pic As PictureBox, hdc As
Long)
Dim x As Single, y As Single
mapMain.FromMapPoint pt, x, y
Dim xPixels As Integer, yPixels As Integer
xPixels = Me.ScaleX(x, vbTwips, vbPixels)
xPixels = Me.ScaleX(y, vbTwips, vbPixels)
BitBlt hdc, xPixels - 16, yPixels - 16, 32, 32, pic, hdc, 0, 0, SRCCOPY
End Sub
```

另一方面，hWnd 是 Map 的真正的句柄。它存在于较大的 hDC 中，尽管它设计成为地图的属性，实际上 hwnd 是 windows 变量，不能够改变，只有运用 API 子程序，才能通过它。

在下面的例子中，当用户在地图上拖动鼠标，就会出现一个矩形，依据矩形的大小显示地图。程序得到了地图的 hwnd 属性，用来设置显示设备。

```
Private Declare Function Rectangle Lib "gdi32" (ByVal hdc As Long, ByVal X1 As
Long, ByVal Y1 As Long, ByVal X2 As Long, ByVal Y2 As Long) As Long
Private Declare Function GetDC Lib "user32" (ByVal hwnd As Long) As Long
Private Declare Function ReleaseDC Lib "user32" (ByVal hwnd As Long, ByVal hdc
As Long) As Long
Private Declare Function SetROP2 Lib "gdi32" (ByVal hdc As Long, ByVal
nDrawMode As Long) As Long
Private Const R2_NOTXORPEN = 10
Private Const R2_NOT = 6
Dim g_hdc As Long
Dim g_hwnd As Long
Dim dragging As Boolean
Dim xs As Integer, ys As Integer
End Sub

Private Sub Form_Load()
dragging = False
```

```
End Sub

Private Sub Map1_MouseDown(Button As Integer, Shift As Integer, X As Single, Y
As Single)
dragging = True
g_hwnd = Map1.hWnd
g_hdc = GetDC(g_hwnd)
SetROP2 g_hdc, R2_NOTXORPEN
xs = Form1.Scale(X, vbTwips, vbPixels)
ys = Form1.Scale(Y, vbTwips, vbPixels)
Rectangle g_hdc, xs - 10, ys - 10, xs + 10, ys + 10
End Sub

Private Sub Map1_MouseMove(Button As Integer, Shift As Integer, X As Single, Y
As Single)
If dragging Then
Rectangle g_hdc, xs - 10, ys - 10, xs + 10, ys + 10
xs = Form1.Scale(X, vbTwips, vbPixels)
ys = Form1.Scale(Y, vbTwips, vbPixels)
Rectangle g_hdc, xs - 10, ys - 10, xs + 10, ys + 10
End If
End Sub

Private Sub Map1_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
Rectangle g_hdc, xs - 10, ys - 10, xs + 10, ys + 10
ReleaseDC g_hwnd, g_hdc
dragging = False
d = Form1.Scale(10, vbTwips, vbPixels)
Dim pt As New MapObjects.Point
Set pt = map1.ToMapPoint(X, Y)
dist = map1.ToMapDistance(d)
Set rect = CreateObject("MapObjects.Rectangle")
rect.Left = pt.X - dist
rect.Top = pt.Y + dist
rect.Right = pt.X + dist
rect.Bottom = pt.Y - dist
map1.Extent = rect
End Sub
```

向层里加数据

在往地图控件中的层集里加入图层和影像层之前，你需把层集与数据相连接。通过 DataConnection 和 GeoDataset 目标可从文件系统或 SDE 数据服务器中查找和连接数据。

DataConrection 目标

DataConrection 目标用来连接装有形文件的文件夹或 SDE 数据库。
要连接形文件文件夹，需设置数据库特性为具有文件夹名的串，并应用 Connect 方法。
连接 SDE 数据库，需设置数据库，密码，服务器和用户特性，并应用 Connect 方法和检查连接特性。
如果连接错误，检查 ConnectError 特性，对比 ConnectionErrorCode，寻找错误原因。
当你进行连接时，GeoDatasets 就会移到当前层或 SDE 层中的形文件集里。GeoDatasets 并不是自动移到层集里，而是目的在于组织一个接口，用户可以选择形文件或 SDE 使用 GeoDatasets 集中的 FindGeoDataset 方法，来使形文件或 SDE 层连到 GeoDatasets
你可使用 AllGeoDataset 方法生成一个新的形文件。当你用这样方法生成一个形文件，也就是生成了一个 TableDesc 来设置新的形文件的生成。请参照 MapObject 手册或 MapObjects 帮助功能中的程序例子。
GeoDataset 目标和 GeoDatasets 集
GeoDataset 表示从形文件或 SDE 层中得到的地图数据的一层。Geo Datasets 集表示 DataConnection 里所有的 GeoDatasets，即文件夹里所有的形文件或 SDE 数据库里所有的 SDE 层。
注意，MapObjects 中 GeoDataset 属性是只写的。一旦你将 GeoDataset 放置到 MapLayer 或其它目标上作为一种属性，那么它不能恢复或改写。
MapObjects 里有 3 种属性：MapLayer 目标的 GeoDatasetct 属性；AddressMatcher 目标的 SrreetTable 属性；PlaceLocator 目标的 PlaceNameTable 属性。
增加形文件
使用 MapObjects 的一个最基本的任务是往你的地图里增加形文件。以下是增加形文件的步骤：

1. 调出一个新的 DataConnection
2. 设置数据库属性为包含形文件的文件夹。
3. 调出一个新的图层。
4. 在 DataConnection 上使用 FileGeoDataset 方法，用形文件名设置 Map Layer 的 GeoDataset 属性。
5. 向层集里加入图层。

```
Dim dConn As New MapObjects.DataConnection
Dim mLayer As New MapObjects.MapLayer
dConn.Database = "c:\MapObjects Data\Namerica\Mexico"
Set mLayer.GeoDataset = dConn.FindGeoDataset("Roads")
Map1.Layers.Add mLayer
```

加入 SDE 层
如果你需要运行一大规模地图数据，可使用 ESRI 的 Spatial Database Engine（空间数据引擎）。注意，必须向服务器中按装 ESRI 的 Spatial Database Engine
加入 SDE 层的许多步骤同加入形文件是一样的，但还需提供服务器名称，用户名称，密码以及 SDE 数据库名称。

```
Dim dCon As New MapObjects.DataConnection
With dCon
.Server = "SpeedyServer"
.User = "SDE user"
.Password = "Top secret"
.Database = "MondoMaps"
```

```
End With
dCon.Connect
Dim SDElayer As New MapObjects.MapLayer
Set SDElayer.GeoDataset = dCon.FindGeoDataset("LayerIWant")
Map1.lapers.ADD SDElayer
```

在 SDE 层中，GeoDataset 集对决定哪些层出现在 SDE 数据库里也是有用的。当你连接上一个 DataConnection 目标，GeoDatasets 集将置于一个指定的 SDE 数据库的 SDE 层里。

在形文件里，通常不用 GeoDatasets 集，因为你将使用 VB 的通用对话方式寻找一个目录下的形文件。而 VB 普通控制不能在 SDE 层里运行，GeoDatasetsconection 能用来建立一个 VisualBasic 控件例如 ListBox，ComboBox 或 ListView

增加一个图象文件

增加一个图象文件是很容易的。你需做的只是申请一个新的 ImageLayer 目标，赋一个名字并把它加到一个地图的层集上去。

以下是一个增加图象文件的简单编码的例子：

```
Dim iLayer As New MapObjects.ImageLayer
iLayer.File = "c:\data\Washington.bmp"
Map.layer.Add iLayer
```

你必须记住，如果 MapObjects 能找到相关的配准文件，它将自动应用这个文件。

TrackingLayer 动态跟踪层操作

你可以在地图上动态地显示一个随时运动的物体。例如，你可以在你的地图上显示救护车，飞机，货车。动态跟踪层最适合接收从 GPS 接收器中接收来的实际的时间空间数据并在地图上动态显示。

TrackingLayer 和 GeoEvent 目标使你能显示随时间运动的模型。

TrackingLayer 和 GeoEvent 目标

TrackingLayer 目标代表一个在你的地图控件中的一层，它显示与层集之后，并可相对层集独立重显。GeoEvent 目标表达一些 TrackingLayer 里的离散目标，这些目标可以用编程的方法移动。

TrackingLayer 和 GeoEvent 目标的属性和方法

当你向一个表单增加地图控件时，你就自动拥有一个 TrackingLayer 目标，它是地图的一个属性。Event 组是空的，EventCount 的值为零。使用 TrackingLayer 里的 AddEvent 方法可生成一个新的 GeoEvent 目标。

```
Map1.TrackingLayer.AddEvent 25000, 45000, 0
```

一旦你加入 GeoEvent，它就被加入 TrackingLayer 目标的 Event 数组中。你可通过分配 TrackingLayer 目标中的 SymbolIndex 给 GeoEvent 目标分配符号。你还可通过编程时时控制符号的位置和方向。

Refresh 方法可以不重显层集而单独重显 TrackingLayer，这是 TrackingLayer 的一个重要的特性，因为你可以不用调用下面的地图层而快速地完成显示。当你调用 Refresh 时地图控件的两个事件将被引发，BeforeTrackingLayerDraw 和 AfterTrackingLayerDraw

为地图层建立一个用户界面

在所举例子的代码中我们已讲过了增加形文件，SDE 层和图像文件的方法。我们编程输入数据资源，而忽视了这些资源的有效性，并且在建立界面时没有用 VB 的其它任何控制。

现在，我们将以 moView 为例，为地图层建立一个用户界面。

介绍 moView
我们将从遍历 MapContents form 所有性能开始，接着我们将解释它代码的性能。
如果你按一下工具栏里的 Map 按钮，MapContents 框将显示。
用 Add File，和 Add SDE Layer 按钮给一个地图增加层。一旦你增加了一些层，moView 的主表单将开始显示地图数据，MapContents 框里的一个 Visual Basic List View 控制将层名以及状态的说明。
MapContents 控制里有八种设定的状态：点图层可见，线图层可见，多边形图层可见，图像层可见，点图层不可见，线图层不可见，多边形图层不可见，图像层不可见。
在表单中查找一个文件
要增加一个图形或图象文件时，击 AddFile 按钮。你将会看到常见的打开文件的对话框出现。注意，形文件(*.shp)为缺省文件类型，但你可以按类型 BOX 来选择两个常用的图象类型 windows bitmap(*.bmp)和 Tagged Image File Format(*.tif)，或者其它任何图象文件类型。
下面是 Map Contents 表单里用来建立常见的对话框和返回文件的编码。这个过程处理以"shp"为扩展名的形文件和其它的图象文件，这些文件增加层的过程中实现。
往表单中加入一个形文件
如果加入文件的过程中发现一个带有"shp"后缀的文件。那么这就是往地图层集中加入文件的过程，注意，由于地图是在不同于地图控件的表单上，层集被定义为.mapDisp.Layers，frmMain 是起动表单，MapDisp 是表单上的地图控件，Layers 是地图的层集。
**********P62
往表格中加入图象文件
如果加入文件的过程中发现不是形文件，那么这就是往地图层集中加入 ImageLayer 的过程。在 moView 上，往层集里加入图象文件时，总是加到了层集底部，以避免它覆盖其它层。
**********P62
往表格中加入 SDE 层
在 MapContents 表单上，按 AddSDELayer 加入 SDE 层。表单会显示一个对话框。首先，输入服务器，用户和密码值，这些会设置为 DataConnection 的属性，按下 Connect，联接到服务器上。联接好了以后，这个 SDE 层将在表格右边的显示框中显示出来，可以加入一个或许多 SDE 层。
更改一个层集里的层的顺序
你在 MapContents 表单中上下移动层的顺序。
你将会在 MapContents 表单的底端发现分别带有向上和向下箭头的两个按钮，当你选择表中顶端的层时，你将会看到向上的箭头将会变模糊并且不能实现，同样，当你选择表中底端的层时，向下的箭头将变模糊并且不能实现。首先，选择好要选的层，按其中的一个按钮来提前或排后它在层集中的位置。
向上和向下箭头的按钮在一个由 Visnal Basic 提供的 Toolbar 控制里实现。
建立层集的一个 Listview
在 moview 应用中，我们选择了 Visual Basic List View 控制来管理层集。还有其它的可用的控制：ListBox 或 ComboBox。但我们认定，List View 控制为最好的建立一个用来管理层的用户界面的方法。因为 ListView 控制有一个 List Item 对象，每一个对象可以用来编程以响应一个层。我们可以配合 Map Objects 层集中的方法来用 Visual Basic List Item 里的方法。

我们想用控制的方法来描述一个层是否可见和层的类型。你可以在界面上看到两个标识，但是对于某一个 List Item 对象只有一个标识是可以实现的。下面是我们使用 List View 控制的技巧：我们有两个可见的状态和四种层的类型(点，线，多边形和图相)
所以我们为每种可能建立了八个 bitmaps，把它们放到一个 Imagelist 中，并把 Imagelist 和 Listview 控制联系起来。每当层集的状态发生变化时，MapContents 里将有一个重建 Listview 控制的调用。下面是重建 ListView 的过程：
**********P66
触发 List View 里的可见状态.
在 Map Contents form List View 控制里，你可以在 checkbox 里按键来触发层的可见状态。下面是 ListView 里触发的操作。
当你按一个 ListView 控制里的键时，ListView 的 MouseDown 事件将启动，当你在一个 ListItem 对象里按键时，ItemClick 事件将启动。
下面是 Map Contents form 激发可见状态的步骤：
在 MouseDown 事件中截断 X 和 Y 坐标 mouseX 和 mouseY 是 form 的特有变量。
**********P67
如果 X 的位置小于 checkbox(250 tmips)的宽度就调用 toggle Checkbox 过程。
**********P67
toggle Checkbox 用 Not 运行的方式来弹出可见特性，ListView 被重建，地图被重新画。
**********P67
为地图属性建立界面
Map Contents form 具有四种地图通用显示属性：背景色，边界类型，取消控制，滚动条。当你点 General 表时，便可设置这些属性。
设置背景色

```
Private Sub pctLightYellow_Click()
frimMain.mapdisp.BackColor = pctLightYellow.BackColor
Edn Sub
```

设置边界类型
边界类型决定是否在地图周围划一条线。0 表示无边界，1 表示设定一个简单的固定的线。

```
Private Sub cboBorderStyle_Click()
Dim status As Integer
status = left (cboBorderStyle.Text, 1)
frmMain.mapDisp.BorderStyle = Statue
End Sub
```

设置滚动条

```
Private Sub chkscrollbars_Click()
If chkscrollbars.Value = 0 Then
frmMain.mapDisp.ScrollBars = False
Else If chkscrollbars.Value = 1 Then
frmMain.mapDisp.ScrollBars = True
End If
End Sub
```

设置绘图取消状态

Cancel Action 功能详细说明了当你按下 Escape 键时会出现的情况，我们已讨论过这一功能，数值为 0，不取消绘图；数值为 1，停止所有图层上的绘制；数值为 2，则停止当前一层面上的绘图。

```
Private Sub cboCancelAction_Click()
Dim status As Integer
status = left (cboBorderStyle.Text, 1)
frmMain.mapDisp.CancelAction = Statue
End Sub
```

第三章

坐标系和几何目标

一但你在应用中引入一张地图，你可就要对其进行一系列的操作。地图控件和 Map Objects 中的一组目标提供了许多工具，可以实现移动，放大或缩小地图的比例，地图漫游或查询等功能。

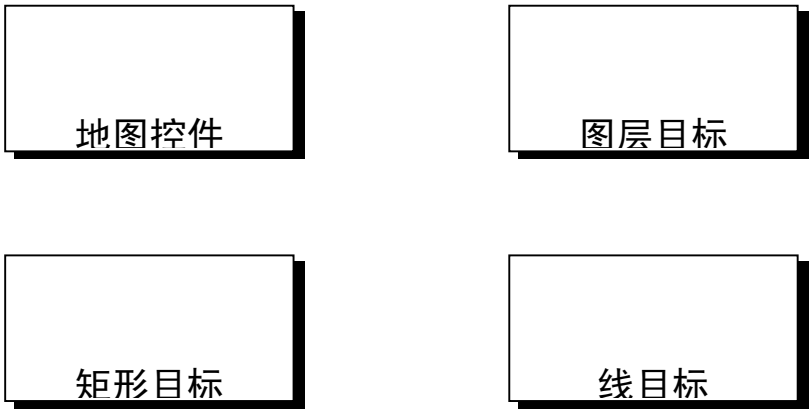
在这一章，我们将探讨一些技巧，来改变正在被观查的地图区域的范围， 计算地图的比例尺，从地图上得到地理坐标，将屏幕上的坐标转换为地图坐标，或是相反操作，选择或添加地物。我们将在 moView 的范例中涉及这些技巧。

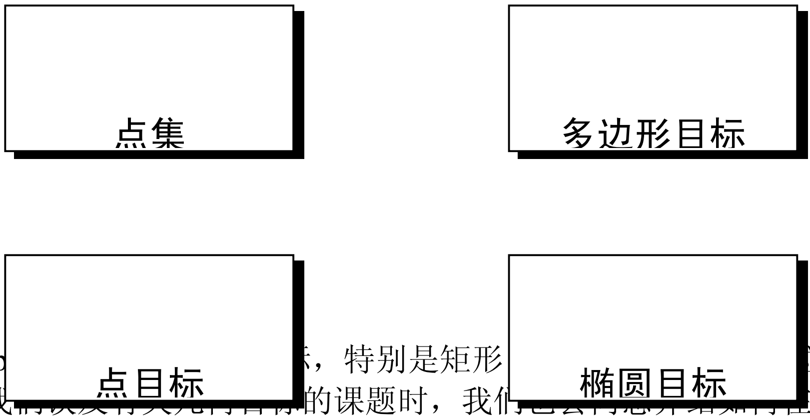
贯穿 MapObjects 终始，我们将常提到几何目标，理解几何目标的关键在于要认识清楚它们是为许多目的服务的。例如：在地图控件中从鼠标交互操作中返回几何目标，返回 所选图形在地图控件中进行点与范围的交互操作。定义被显示的地理区域，创建新图形几何目标等等。

以下是本章涉及的主要内容：

- 控制坐标和表单坐标是如何在屏幕上定义的。
- 使用地理坐标来设定某一地理元素。
- 地理坐标和控制坐标的转换。
- 设置地图范围和对地图比例尺的操作。
- 从地图上得到某一位置。
- 漫游和缩放。
- 重置地图尺寸。
- 从图形上标定几何目标。
- 创建新几何目标。

下面是这一章中我们将讨论的 Map Objects 目标：





Map[illegible]标，特别是矩形[illegible]位置交互操作的关键。当我[illegible]的课题时，我[illegible]地图上画上些基本的几何图形。例如:直线，矩形和圆。

坐标系

当你使用地图控件来显示一个地理区域时，了解 Map Objects 的地理坐标和 Visual Basic 的表单坐标之间的关系至关重要。

控制坐标(control coordinate)

在 Visual Basic 表单中左上角为原点，水平方向为 X 轴，竖直方向为 Y 轴。

Visual Basic 以 twips 作为缺省的测量单位。你可以把表单的单位改成点，pixels，字符长，英寸，毫米，厘米。或通过设置 Scale Mode 属性来实现自定义单位设置。但在我们的讨论中我们应用 twips，因为它是系统默认的缺省单位。在表单内的地图控件也有它自己的坐标系。其坐标单位与表单坐标相同。我们地图控件中的坐标称为"控制坐标"(control coordinate)

地图坐标(Map coordinates)

地图控件也可工作于地理坐标中。遵循笛卡尔坐标系原则。不同于 Visual Basic 表单的坐标系（左上角永远是 0,0）地图控件内显示的地图坐标范围，时常在应用期间改变。 每次当你移动某一地图区时，地图控件内地图坐标范围就会变化。

表单坐标与地图坐标间有一些关键的区别：

- 地图控件的控制坐标的左上角的一位置为(0，0)。地图坐标通常都有一个在地图控件区域很远以外的原点(origin)。应当记住，你的地图控件只是地图表面的一个小窗口。
- 控制坐标 Y 轴延向下递增，而地图坐标 Y 轴延向上递增。
- 控制坐标以 twips 为单位，并且与你的计算机屏幕显示的实际尺寸有关。地图坐标则用米，英尺等单位表示，并且与地表特征的测量有关。

你可以把地图控件当作一个放大镜，在一块地域内可随意移动这个"放大镜"并增大或减小其放大倍数。

你地图数据中所用的坐标值通常很大，有的甚至是成百上千或数以百万计的。这些坐标值通常是以英尺或米为单位的。你地图资料的坐标，可以不同的坐标系为基础，如平面坐标系(SPCS:State Plane Coordinate System) 或麦卡脱坐标系(UTM:Universal Transverse Mercator)以及其它坐标系。如果你是从别人那里得到的资料，那么其提供者应已设置了适当的坐标系。

关于坐标系更细致的讨论超出本书的范围。但是值得注意的一点是应用 Map Objects 时要确认图层上的资料用统一坐标系。如果不是，那些用了不同坐标系的图层是无法连接的。

如果你的地图资料应用了不同坐标系，并且你想同时在 Map Objects 中使用它们，那么你可以用一些其它软件，如"Arc-View"或"ARC/INFO" 来把你的地图资料转换成普通坐标系。该过程称为"map projection"(地图投影法)

地图与屏幕间的坐标转换

地图控件有四种方法将位置和线性尺寸在"地图坐标"与"控制坐标"间转换。

＊＊＊＊＊＊

a: 用 Map Distance 方式，将以地图单位计算的距离长度转换成表格单位制下的距离长度。用 Map Point 方式地图上点的坐标。转换成 地图控件 下标准单位制的 X，Y 坐标。

b: To Map Distance 方式，将以表格单位下的距离长度转换成地图单位下的距离长度。 To Map Doint 方式将 地图控件 下，以表格单位(X，Y) 为坐标的位置转换成为地图单位的一个点。

c: 这些鼠标操作，将返回地图上表格单位下的位置， 您可用这些操作来得到屏幕位置，并且坐标转换方式把它们转换成为地图坐标。

地图控件 下的四种转换方式:

＊＊＊＊＊＊＊＊＊＊＊＊

要得到一点的地图坐标，可用 ToMapPoint 方法。你可使用鼠标事件(Mouse Down ， Mouse Move ， Mouse Up)，从屏幕上选定一点，返回的 X，Y 值就是其控制坐标。

要把地图坐标转换成控制坐标，用 FromMapPoint 方法，你将输入一个点目标并得到控制坐标的 X，Y 值，你可以其它方法应用这些 X，Y 值，来定义位置或选择地物。

若把控制单位下的长度转换成地图单位下的长度，要应用 ToMapDistarce 方法。

把地图单位下的长度转换成控制单位的长度，要用 FromMapDistance 方发。在你的实际应用中，你可用地图控件下的 TrackLine 方法来画出一个轮廓，并得到其长度。

＊＊＊＊＊＊＊＊＊＊＊＊＊＊

P77 (浏览器)

a: "moView 应用"使用 To Map Point 方式以当前地图坐标来记录统计栏内方格控制板上的文本。

以当前光标可移滑块位置的地图坐标。来记录新近文本。

moView 范例包括一个将控制坐标转换为地图坐标的举例：当你在地图图层中移动光标时你将看到当前地图坐标显示在表单底部状态条的第二栏中。

在这个例子中。sbrstatus 是一个在 moView 主栏中的状态条绦记录有地图坐标的文本信息在这状态条纹的第二个方框中被显示。

＊＊＊＊＊＊＊＊＊＊

位置和地图比例尺

地图上最常用的一个操作是改变地图位置和比例。我们现在讨论一下用 map extent 来定位地图，用 map scale 来改变大小的技巧。

在地图控件中地图范围以 Extent 属性代表，Extent 属性是一个矩形目标，通过上，下，左，右边界尺寸属性来定位。这些尺寸是地图坐标数据。

你可用许多方法设置地图的范围。如从某个图层上来取得地图范围或对另外一地图范围进行某些数学操作，你可将地图控件的 Extent 属性设置成你想要的新范围，在 Map Objects 中还有很多方法让你设置地图的范围。

＊＊＊＊＊＊＊＊＊＊＊＊＊＊＊

下面是一些图框和功能你可能会用到。

a: Center At 方式可将图幅移至以(X，Y)为中心的位置。

pan 方式通过拖拉操作。将图幅移动位置。

Track Rectargle 方式。可用来从图上拖出一个矩形框来放大。

b; Interset 方式。可找到两矩形框的交叉位置。

Union 方式。可找到两矩形框的接合处。

Scale Rectangle 方式。增大或减小矩形的尺寸。

c: Extent 是当前图的范围。 Full Extent 是所有图层中最大范围。

d: Map layer 的范围是一个分隔和图形文件或 SDE 图层中所有特征的矩形。

e: Image Layer 的范围是个界定影像文件中所有间隔的矩形。

f: Top (顶) Left (左)　Bettem(底) Righl(右)定成了一个矩形区域。

P79 地图控件 有一项功能叫做 Extent，这是一个矩形目标。

为了考查 地图控件 中 Map1 的地图范围。你可以考查这四项指标。

就象这样:

你可通过中间矩形目标(子窗口)来检查其范围。就象这样。

下面是 Visual Basic 中一些设置地图范围的方法举例。

1 欲设置地图范围成一个分离坐标域(即分别赋值给中项指标)

2 欲将 地图控件 设置为图层集合中第三层的范围。

3 欲将 地图控件 设置为所有图层的最大范围。

4 用一比例系数将 地图控件 设置成放大(200m in)的矩形。

5 将 地图控件 幅面设置为两图层跨越的区域。 加一更清楚但略繁琐的表达方式为:

6 将 地图控件 的幅画设置成两图层交叉处的区域。

另一种清楚但繁琐的书写方式为:

7 欲在操作中得到整幅地图的全景。

当你启用 Pan 方式时，焦点箭头直接指向 地图控件，你的游标可随意设置地图范围。移动地图内的游标(拖住游标)你会看到地图移动。当你松开手时，地图的其它部分就会被重画。

P80

8 以(55000，65000)为中心重新绘制地图:

＊＊＊＊＊＊＊＊＊＊＊＊

你还将在 Map Objects 下发现其它设置地图范围的方法。

一个一般的技巧是让地图范围定义为矩形目标。使用矩形方法来改变矩形目标的尺寸。然后将修改过的矩形目标以地图控件范围大小放回。

计算地图比例尺

地图比例尺是地图单位下的实地尺寸和屏幕单位下的地图控件尺寸之间的比例。下面是一个使用 moView 例子来计算当前图面比例尺：

＊＊＊＊＊＊＊＊＊＊＊＊

计算地图比例尺的举例:

＊＊＊＊＊＊＊＊＊

当你做地图比例尺计算时，使用 X 轴或 Y 轴的值会得到同样的答案，如果你正在做手工运算。执行并比较一下不同尺寸下计算的结果。是一种好的检查方法。

＊＊＊＊＊＊＊＊＊

P81 下面是一个在 地图控件 下计算当前地图比例尺的程序。

在这个程序中，地图的比例尺将按水平坐标轴计算。它也能重过，纵坐标轴来计算。其结果将显示在"浏览器"(moView application)主栏底部的一个窗口内

＊＊＊＊＊＊＊＊

重置地图尺寸

许多 Windows 应用程序允许用户重置窗口尺寸。当使用者重置主窗口尺寸时，只需拖动窗口边框，工作区便会按比例放大或缩小到新的窗口的范围。

你可以实践一下，重置包括地图控件在内的图框的尺寸。这是一幅关于重置浏览器窗口尺寸的示意图。

＊＊＊＊＊＊＊＊＊

a:当你重置 moView application 主窗口尺寸时。。。。。

b:不仅外框尺寸改变了。而且地图控制器(地图控件)的尺寸也随之改变了

伸展外框的同时，也伸展了地图。

P82 这一对 地图控件 需于伸缩性的操作。其关键是对 地图控件 所存在于的外框进行。尺寸重置操作。使用者任何时刻可使用鼠标拖动边框来设置其尺寸。

下面是在 moView application 中如何完成地图尺寸的设置。map Disp 是地图控件 在这一表格窗口下的名字。

＊＊＊＊＊＊＊＊＊＊＊＊＊

这一程序考虑了主菜单下工具栏和状态栏的尺寸，更新 Top，Left，Height 和 Width 属性值，适时的填入表格的剩余空间。这一程序也确保在你将表格重置为小尺寸时。地图控件不会作为是不认可尺寸处理，这在 Visual Basic 中将会造成 run time 错误。

当你运行 moView 并设置图框尺寸时，你也会看到地图比例尺随之变化。这一现象发生在图层绘制操作过程中，之所以如此是因为重置地图控件的尺寸，系统会自动启动 Refcesh 方法。Before Layer Draw 这一指令下达到一个叫做 Update scale 的程序中，这一程序可计算新图的比例尺，并将所得结果显示于状态栏中。

注意到地图的范围在一程序操作中并未改变，所以所显示的地理区域保持不变，只是地图的比例尺规模发生了变化。

如果你需要，你可加一些指令在程序中，来保持地图比例不变，并缩小或放大地图的范围。你可使用计算和设置合适的地图范围的方法。来保证比例不变。

建立一个漫游和缩放操作的工具条

工具条是一个下达命令的自然友好界面控制器，因为它使这些常用命令易于执行。＊＊＊＊＊＊＊＊＊＊＊＊＊

－＊＊＊＊＊＊＊＊＊

主控制板上的工具条有四个关于漫游和缩放的按键。

Full Extent 改变地图范围至图层集合中该图的全部范围，这一命令将地图控件的 Extent 属性设置到了 Full Extent 属性。

Zoom in 激活了地图控件中的 Track Rectangle 方法。当你在地图控件中移动光标时，你可拖住一个矩形框，而当你放开时，地图的新范围就会被设置在该矩形框中。

Zoom out 按照一个比例系数，以你在图上选定的点为中心计算新图的范围

Pan 激活了地图控件的 Pan 方法。在地图控件中拖住光标来移动地图使其以新的一点为中心。

Zoom in，Zoom out 和 Pan。属一个按键组，在这一按键组中，也包含了 Identity 和 Graphics 按键。这五个按键的共同之处是他们都是在地图控件中鼠标操作状态下的。例如：当按下 Zoom in 时，其它按键则被关闭，当你在图内移动光标 Mouse Down 会为 Zoom in 操作执行 Track Rectangle 方法，而不是为别的按钮服务。Full extent 命令则不是该按键组的一部分。因为它不需鼠标操作而立即执行。

这些是应用工具条中命令使用的大体步骤。我们将举例来应用这些浏览器中控制器和指令。＊＊＊＊＊＊＊＊＊＊＊＊

1 为了你实践需要，建立小块地图。标准为 16*16 Paint 或 rester (光栅)编辑程序来建立地图。

P84 在 Paint 中编辑一个放大的 16*16 pixel 的地图块。

2 将 Image List 控制器加到你的控制极上，如果你地图的颜色不合适，到了 Image List 功能的色彩表中将 Mask Coloz 功能设置到未曾使用在该地图上的颜色。

在浏览器中检查一个 Image List 的控制器。

3 将一个工具栏(Tool bar) 控制器加到主控板上并将"查照"加到 Image List 控制器上。接下来借助于在 Image List 控制器内的检索来增加关于图象的按键。工具栏功能菜单使得作为按键组和工具 tip 更易于提供图特征图例。除非你想要一幅大图象并在下面标注。

插图说明(或字幕)，否则应避免设置 Caption 功能。

通过 Visual Basic4，改进 Tool Bar 和相关的 Image List 是很麻烦的事。为了使 Image List 控制器和新图保持同步。你必须首先从 Tool bar 控制中分离出 Image List

P85 (然后 Tool Bar 中的所有按键将变为横线)，接下来， Mpolate 你的 Image List 并将其一重新分配到你的 Tool Bar 控制器中，然后，浏览所有按键， 并重新输入图象检索值。遗憾的是，这是应用在 Visual Basic 中使 Tool Bar 和 Image List 控制器成为适时状态的程序。

4 将指令加入到 Mouse Down 操作中使系统执行"全景"操作和"远近"操作命令。下面是一些在浏览器中 地图控件 上进行 Mouse Down 操作的指令 map Disp 是这个 Map conrtol 的名字。bar Disp lay 是含有 pan， zoom

命令的一个(Tool bar control)工具栏控制器的名字。

与识别命令和几何图形绘制相关;更多指令，在这一操作程序中紧随其后。

＊＊＊＊＊＊＊＊＊＊＊＊＊

几何目标

到现在为止，我们还未明确讨论过几何图形。但我们已经用过它来进行地图范围的变换操作和用鼠标事件从地图控件中返回点。下面是 Map Objects 中由六种几何目标形成的图形。＊＊＊＊＊＊＊＊＊

＊＊＊＊＊＊＊

所有为几何目标定义的坐标信息都是地图单位制，用地图控件中的 FromMapPoint 和 ToMapPoint 方法把控制坐标转换成地图坐标。

Map Objects 中的几何目标具特色。你可用它们来返回在地图控件上的位置和图形。从图层中获得所选取的图形，控制地图控件中地图的范围。或者检验某一特定目标是否存在于其它的目标之中。

地图控件返回几何目标的四种方法是：Track Line，Track Rectargle，Track Cirtle，和 Track Polygon 这些事件可用作选择，测量，生成图形和其它目的。

图形与几何目标

六种几何目标中，点，直线和多边形有相同的几何定义。与形文件中和 SDE 图层中相应的特征形一样。

形文件和 SDE 图层不能直接贮存椭圆和矩形。但它们可将其近似为多边形。下面是如何通过存在于形文件和 SDE 图层中的特征图例来绘制几何目标。

＊＊＊＊＊＊＊＊＊＊＊

P87 ?a 从一个名为 "shape" 的 Recordset 中返回 Field 目标中的 Value 功能。通过这一操作。你可以图形文件或 SDE 图层的图例中建点，线或多边形目标。

b 从图形文件中或 SDE 图层上返回的"直线"和"多边形"目标有阴影部分。凭借 Recordset。你可以从图形文件或 SDE 图层的图形中创立几何图形。下面是一些通过 Recordset 重重显示每个图形的简单指令。

假定你有一个叫做 Map1 的 地图控件 和一个含有多边形叫做。

Poly Layer 的 Map Layer (图层)

(相关于)

注意:有可能将图形写到图形文件中这并不符合关于图形。文件的详释。 一个例子是。修改一个多边形的点集(Points colletoom)以使首尾两点不同。

在几何图形上可进行的关于 Map Objects 的操作，对图形文件中的图例来讲可能是不合法的。 Map Objects 不在你写到图形文件中的图形上。 执行任何合法操作。当读到你用 Map Objects 错误的建立或编辑的图形文件时， 你可能会碰到一些反常的现象。或软件运行失败。

P88 直线与多边形允许多部复合。

与图形文件和 SDE 图层信息中的相应特征类似。 直线与多边形图形都能支持组合图形。意即直线(Line objects) 能构成不连续的折线而这些不连续的折线构成拱形(Polygon obj)多边形。能构成许多环节，这些环节形成了一个多边形文件。

直线与多边形图形能代表与图形文件和 SDE 图层兼容的组合图形，但它们的方式与性能并非为控制处理组合图形而设计。在 Map Objects 中建立组合直线与组合多边形的唯一途径是从(内存) Recordset 中调用一图形来设置一条直线或一个多边形。同样也无法 programmatically 发现。某一给定图形是否由单体或多部分多合而成。你能在 Map Objects 中对简单图形进行简单编辑， 但修改编辑却是其它软件诸如 Arc View 或 ARC/INFO 之类的事。 Map Objects 不支持组合点图形文件。

＊＊＊＊＊＊＊＊＊＊＊＊＊＊

点目标

点目标代表一种在地图坐标中具有 X，Y 值的几何图形。点的一般用途是：

- 代表点形文件或 SDE 图层中的点。
- 返回与相关位置相匹配的地址。
- 代表地图控件上由使用者敲击鼠标的点。
- 在直线和多边形目标中返回和设定坐标。
- 返回几何图形交叉处的坐标。

/*******************

P89 A X，Y 是地图坐标系统中的迪卡尔坐标。

B Distance To 方式计算两点间距。

C Distance To Segment 方式计算点与由两点连成的直线之间的距离。

Peint Object 的性能与用法。

＊＊＊＊＊＊＊＊＊

以下是返回 Point 目标的属性与方法：

- 矩形与椭圆目标的中心属性。
- 多边形几何重心属性。
- 点集的 Item 方法。
- Address location 目标的定位(Loction)属性。
- 地址匹配 (Address Matcher)目标的 Match Intersection(交叉点匹配)方法。
- 地图控件上的 To Map Point 方法。
- 字段(Field)目标的值(Value)属性。

以下是设置及通过 Point 目标的属性与方法：

- 点集的增加(Add)方法。
- 点，线，多边形与矩形目标的距离到(Distance To)方法。
- 点目标的 Distance To segment 方法。
- 地图控件的 Draw shape 方法。
- 地图控件的 Draw Text 方法。
- 地图控件的 Flash shape 方法。
- 地图控件的 From Map Point 方法。
- 点，线，多边形，和矩形的猎取交叉点方法(Get crossings)
- 点集的插入方法。
- 多边形，矩形，和椭圆形内 Is Point In (点是否存在)方法。
- Map Layer 目标的 Search By Distance 方法。
- Map Layer 目标的 Search shape 方法。
- 点集合的 set 方法。
- Field 目标的 Value 的属性。

点集。

点集(points collection)由一组点目标组成，其目的是代表线和多边形目标的几何形状。下面是一部分点集的常用操作：

- 编辑线和多边形目标的几何形状。
- 返回重叠几何目标的交叉点。
- 改变线和多边形的方向。

＊＊＊＊＊＊＊＊＊＊＊＊＊＊

＊＊＊＊＊＊＊＊＊＊＊＊＊

点集是不可生成的目标。可把点集作为线或多边形等目标的一个确属性进行编程，或可利用某种方法。下面是更新线目标点集的例子。

＊＊＊＊＊＊＊＊＊＊

＊＊＊＊＊＊＊＊＊＊

以下是返回点集的属性(property) 和方法(methods)

- Get Crossings 方法(点，线，多边形，矩形目标)
- Place Locator 目标的 locate 方法。
- 线和多边形目标的点属性。

Add，Insert，Remove，Reverse，Set(增加，插入，转移，反转，设定)是修饰点集的点集方法。

矩形目标

矩形目标在 Map objects 中有广泛的应用，并对地图范围管理有重要作用。下列是矩形目标的常见用法：

- 返回，修改地图范围。
- 代表其他几何目标的范围。
- 返回地图控件上鼠标拖拽的区域。
- 代表图层中形的坐标区域或影像层的单元坐标区域。
- 在矩形区域内选择其他目标。
- 在地图控件上绘出几何形状。

矩形(Rectaingle) 是可生成的目标。下面是生成新矩形目标的例子。

＊＊＊＊＊＊＊＊＊＊＊＊＊

P92　　图形见原材料

－＊＊－＊＊＊＊＊＊＊

以下为返回矩形目标的属性和方法：

- Map Layer 目标的 Area Of Interest 属性。
- 地图控件，图层，影像，椭圆和多边形目标的范围属性。
- 地图控件的全范围属性。
- 地图控件的 Tack Rectangle 方法。
- 字段目标的值属性。

下面是设置或通过矩形目标的性质和方法：

Distance To 方法(点，线，多边形和矩形目标)

- 地图控件的 Draw shape 方法。
- 地图控件的 Draw Text 方法。
- 点，线，多边形和矩形目标的 Get Crossings 方法。

- 矩形目标的 Intersect 方法。
- 矩形目标的 Intersects 方法。
- 矩形目标的 Union 方法。
- Map Layer 目标的 Search By Distance 方法。
- Map Layer 目标的 Search shape 方法。
- Field 目标的值属性

Inset，Intersect，Offset，Scale Rectangle，Union 是修改矩形目标的方法。

线目标

线目标是具有线性几何特征，其几何形状由点集描述，点集中至少有两个点，下面是线目标常用法。

- 代表线形文件(Line shapefile)或 SDE 层次的形状特性。
- 返回用户用鼠标定义的条线。
- 在地图控件上画出几何形状。
- 选出与线交叉的其他目标。

在前面讨论过，线目标可以有很多部分这种情况用来代表弧段。

****

图表见材料

线是可生成目标。下面是应用 Track Line 方法生成一条新线的简易代码。

程序

****

下面是返回线目标的属性和方法：

- 地图控件上的 Track Line 方法。
- Field 目标的值属性。

下面是设定或通过线目标的属性和方法：

- 点，线，多边形和矩形目标的 Distance To 方法。
- 点，线，多边形和矩形目标的 Get crossings 方法。
- 地图控件的 Draw shape 方法。
- 地图控件的 Draw Text 方法。
- Map Layer 目标的 Search By Distance 方法。
- 地图控件的 Flash Shape 方法。
- Map Layer 目标的 Search Shape 方法。
- Field 目标的值属性。

多边形目标

多边形目标具有线性几何特性，可构成一封闭图形。它的几何形状由点集描述，点集中至少要有三个点。

下面是多边形目标的常见用法。

- 在形文件或 SDE 层中代表地物。
- 返回用户用鼠标定义的多边形。
- 在地图控件上绘出几何形状。
- 选择与多边形交叉的其他目标。

****

多边形是可生成目标。下面是根据指定的坐标生成一新多边形的程序

注意在上面的例子中多边形的起点和终点坐标一致。在一多边形目标的点集中加入点目标时， Map Objects 不可能。

P95 辨别哪个点是最后一个，因此不能证实多边形是否闭合。

因此用户就要写下代码确保多边形目标的关闭。层管多边形不闭合时不会出现运行时错误，但是区域和图形不会按预期的样子给出。

下面是另一段代码片段，可以根据用户的反应去 Map 控制上返回一新多边形。

使用上述代码可以保证多边形是闭合的。此类代码会被很典型地放置在 Map 控制的 Monse Dewn 事件中。

图表见材料

****

下面是能够返回多边形目标的属性和方法：

- 地图控件的 Track Polygon 方法。
- Field 目标的值属性。

下面是设定或通过多边形目标的属性和方法。

- 点，线，多边形和矩形目标的 Distance To 方法。
- 点，线，多边形和矩形目标的 Get crossings 方法。
- 地图控件的 Draw shape 方法。
- Map Layer 目标的 Search By Distance 方法。
- 地图控件的 Flash Shape 方法。
- Map Layer 目标的 Search Shape 方法。
- Field 目标的值属性。

一种修改多边形目标的方法是 Offset

椭圆目标

椭圆目标是一个表示椭圆和圆的几何体。它不常在 Map objects 中用到。

以下是椭圆目标的一些用法：

- 在地图控件上画圆或椭圆。
- 在地图控件上得到用户定义的圆。

****

Inset 法:缩小椭圆。 Top，Boffom，Left，Right:

Is Point In 法:测试一点是否在椭圆内。 设置椭圆大小。

Offset 法:移动椭圆。

Ellipse object

Center: 椭圆中心点。

Extent: 椭圆的矩形外界线。

Height，Width: 椭圆短，长轴长度。

图 3 椭圆对象

以下程序代码完成的工作是:在 地图控件 上得到一椭圆对象。

```
Dim cur Ell as New Map Objects    Ellipst
Set cur Ell = Map1    Track Circle
```

圆可看作是椭圆的一个特例，创建一个圆对象时，长轴和短轴应设置为相同。此代码将会在 地图控件 的单击事件中执行。

****
返回一个椭圆目标的方法是：
- 地图控件上的 Track Circle 方法。

通过一个椭圆目标的方法是：
地图控件上的 Draw Shape 方法。
两种修改椭圆实体的方法：Insert 方法，Offset 方法。

第四章

图属性的符号化
本章，我们将探讨符号化图层和用带属性值符号描述地物两种操作的技巧。如果你愿意使你的图如同一幅艺术画，那么精通 Symbol 和 Renderer 是非常是有必要的。
本章包括以下几个部分：
- 用符号表示地物。
- 确定 Visual Basic 和 Map Objects 颜色属性。
- 使用 Map Objects 中颜色，标记，线型，填充类型常量。
- 用特殊的属性值符号化地物。
- 用 True Type 字体注记。
- 分类画地物。
- 用点密度描述多边形中的属性。
- 按照属性标注文本。
- 用符号画用户定义的图形。
- 在一点或直线所确定的位置上写用户定义的文本。
- 打印图形。

为了帮助你开始使用 Symbol，Text Symbol 及 Renderer object，我们来看这几个选自应用实例的范例，并通过这几个例子来说明 Symbol，Text Symbol 及 Renderer object 的使用。
Symbol 目标制定颜色，厚度和类型属性，Text Symbol 目标制定字型及其他文本属性。DotDensityRenderer，ClassBreaksRenderer，ValueMapRenderer 和 LabelRenderer 目标提供依照属性值画地物的方法。
地图元素的符号化
当你往图形上增加图层时，你会很快注意到系统已预先给各层着以不同的颜色。这是为了便于区分各层，但通常，你要控制为各层定义的符号。
Symbol 目标
MapLayer object 的 Symbol 属性决定一个图层或几何对象中以何种样式画出。
你可以读写 Symbol object 中所有属性。然而有一些属性是否可用取决于 Symbol Type (符号类型) property 是否被设置为 moPiontSymbol，moLineSymbol，moFillSymbol 中的一个常量。
当你创建一个新的 MapLayer object 时,它会自动先给 Symbol Type 设置一缺省值：Geo Datasets 中的点状地物置为 moPiontSymbol，GeoDatasets 中的线状地物置为 moLineSymbol，GeoDatasets 中的多边形地物置为 moFillSymbol

你可直接配置 SymbolType 属性而不必考虑符号的实现。例如，在图层中，不必将 SymbolType 置为 moPiontSymbol 而直接符号化多边形中心。
以下五个目标均与 Symbol 目标有关：

- 图层目标有 Symbol 属性，用统一的符号来画一图层上所有属性，这是 Symbol object 的最一般用法。
- 地图控件的 DrawShape 方法是使用 Symbol object 来画几何目标
- ClassBreaksRenderer object 用一组 Symbol objects 来画各类地物。
- ValueMapRenderer object 是把一组 Symbol objects 赋给几个具有特定属性值的地物。
- TrackingLayer object 是用一组 Symbol objects 来画 Geo Event 目标。

颜色属性
MapObjects 中有关颜色的属性沿用 VisualBasic 中各种颜色的定义，也就是一种颜色是一长整型整数，其各字节表示红，绿，蓝的组成。你有许多确定颜色的方法：长整型整数; Visual Basic 中的颜色常数如
vb Blue; Visual Basic 中 RGB 或 QBColor 函数;或 Map Objects 的颜色常数如 moRed.
Map Objects 为方便使用而提供了 23 个颜色常数,这样便易于提供 Visual Basic 中没有定义的颜色.详情请查阅 Map Objects Programmer's Reference 中各种颜色的长整形整数值以及与颜色确定有关的细节.

mo Black	黑	mo Yellow	黄
mo Red	红	mo Lime Green	灰绿
mo Green	绿	mo Teal	
mo Blue	蓝	mo Dark Green	墨绿
mo Magenta	紫红	mo Maroon	紫酱
mo Cyan	氰蓝	mo Pur Ple	紫
mo White	白	mo Orange	橙
mo Light Gray	浅灰	mo Khaki	土黄
mo Dark Gray	深灰	mo Olive	橄榄绿
mo Gray	灰	mo Broun	褐
mo Pale Yellow	暗黄	mo Navy	天蓝
mo Light Yellow	浅黄		

标记,线和填充方式
把 Symbol Type 属性设定为 mo point Symbol mo Line Symbol 或 mo Fill Symbol 三个常值之中一个,也就是, 取一组常值集合中一个常值给 Symbol object.

mo Point Symbol 点	符号类型 有 3 个常数.每个符号类型都
mo Line Symbol 线	对应有一组用于符号对象的符号式样常值
mo Fill Symbol 填充	

mo Circle Marker	圆形点	mo Solid Line	实线
mo Square Marker	方形点	mo Dash Line	虚线
mo Triangle Marker	三角形点	mo Dot Line	点虚线
mo Cross Marker	十字形点	mo Dash Dot Line	点划线
mo True Type Marker	True Type 点	mo Dash Dot Dot Line	双点划线

点的式样常值　　线的式样常值　　填充的式样常值

mo folid Fill　　实填充
mo Transparent Fill　　空填充
mo Horizontal Fill　　水平线填充
mo Vertical Fill　　垂直线填充
mo Vpwarkd Diagonal Fill 左上斜线填充
mo Dounward diagonal Fill 右上斜线填充

mo Cross Fill　　网状填充
mo Diagoral Cross Fill　斜网状填充
mo Light Gray Fill　　浅灰填充
mo Gray Fill　　灰填充
mo Dark Gray Fill　　深灰填充

如果将 Symbol Style 常值设为 mo True Type ,Symbol object 会用 Font 和 Character Index 属性以及 True Type 字体来写字符.若需要定制标志符,可创建一新的 True Type 字体来得到新符号.

Map Objects CD-ROW 包括:几种 True Type 字体和很多可供使用的标志符.这些制图字体需在安装 MapObjects 时选择安装

mo View 应用程序中符号化图层

现在,我们将要探讨在 mo View 应用程序中符号属性是怎样通过表单和代码以对话的方式应用于一终端用户的.

在 mo View Visual Basic 项目中,有与所见表单相关的表单文件 Main.frm,Map Contents.frm,Layer Prop.frm 和 Symbol Prop.frm.其中 Symbol Prop.frm 与 draw Symbol. cls 类模块相关联. draw Symbol.cls 类模块完成的工作是:将当前 Symbol object 显示在 Map Control 上.

Moview 中的符号属性表

mo View 中的符号属性表的功能是:更改 Symbol object 的所有属性.

Symbol Prop.frm 中的 Mpdata Form By Symbol 过程反映了 SymbolProperty form 如何随着符号类型的不同而不同的.

符号属性表单左上角绘有一符号.这个符号的象素信息由 draw Symbol 类模块写入 Map 控件 中. Symbol Prop.frm 中的一个过程调用了 draw Symbol,这个 draw Symbol 类用到 Map 控件首先,计算出 Map 控件的中心;其次,画出一个符合当前属性的符号;最后,这个符号以一窗口文件的形式返回.

图元文件的作用是,该过程把该图形化符号划分为一个个信元放于-Grid 控件中以达到创建一个交互式图例的目的. Symbol Property form 不用 Grid 控件而 Layer Property form 要用到 Grid 控件.实际应用中,这也是一种建立生动符号图例的技术.

用 Draw Shape 和 Draw Text 两种方法在 Map 控件上画图例, 比如填充柜形状,描述性文本以及各种符号.

Symbol Property form 中式样的列表随着当前 Symbol Type 不同而升级.

Layer Prop.frm 中的 load Style Combo 过程反映了 Combo box 更新情况.

在单击符号颜色上(标签 color: 的旁边)的 Picture Box 控件之后被激活的过程将调用普通对话框控件进行颜色的选择.

tempSymbol 是保存当前符号对象的全局对象,以供 Layer Prop.frm 使用.draw Symbol 是一过程,它调用 draw Symbol 类在图象控件上画一指定的符号.

Renderer object

尽管控制层显示属性最简单的方法就是给一图层设置符号属性, 但是其局限之处就是:在一个图层上所有属性都用同一符号描绘.在这一点上, Renderrt Objects 提供了更大的灵活性,可用带参数的数据描绘属性.

Map Objects 中有四个 Renderer objects:

Value Map Renderer object :把一组符号赋给几个唯一值来画元素.

Class Breaks Renderer object 把一组符号赋给一定值域的几个值来画元素.

Dot Density Renderer object 以不同密度的点状图块对应不同的值的方式画多边形.

Label Rendrer object 从属性值下一个元素中提取文本.

在以下几个部分,我们将探讨 Renderer objects 的一些实际应用.下面是适用于 Renderer objects Symbol objects, Text Symbol objects 的几条基本规则:

用唯一的属性值符号化地形要素

你经常会用唯一属性值符号化地图.假设当前你有一个图层,由几块陆地组成:一些是住宅区,一些是公园,一些是工业区等等.现在要干的工作是,从 Recordset 字段中提取属性按土地使用类别给土地使用图着色.

Value Map Renderer object 就是用来生成这类图的.

Value Map Renderer object

在 Value Map Renderer object 的应用中, 若事先并不知道唯一的属性值是什么,通常在 Map Objects 中收集唯一的属性值的方法是: 迭代设置记录和使用字符串数组.如果一开始便设置字符串 Vnique property (唯一属性)为真,那么仅仅当一字符串原先不在字符串数组中时,这个字符串才能用增加法(Addmethod)来增加入该字符串数组.Count properyt(计数属性)表示字符串收集中唯一的属性值即不重复属性值的个数.值数组中没必要包括翻译区域中每个唯一的属性值.例如:共有 20 种土地使用类型,为了只给其中五种主要的土地使用类型着色而只用一种符号描述剩下 15 种土地使用类型,所要做的工作是:给值数组和符号数组设置 5 组值;设置 vse Default 属性为真;设置该图层的符号属性就是剩下 15 种土地使用类型的描述符号.

mo View 中 Value Map Renderer 的使用

mo View 应用程序的 Layer Property form 为 Value Map Renderer 提供了一个对话框.以下是在 mo View 应用程序中创建一个值图的步骤:

1. 主菜单中单击 Map Contents 图标.
2. 在 Map Contents form 中增加一个图层.
3. 单击 Map Contents form 中的 properties.,出现 Layer property form 后,单击 Value Map.
4.选择所感兴趣的区域.应用程序或许花费几秒钟的时间来计算(得到)唯一属性值的列表.如果计算出唯一属性值超过 50 条, 就会出现一消息框询问是否要查找更多的唯一值.此时,最好单击消息框中的 NO ,然后选择下一个具有较少唯一属性值的区域.
5. 由你决定是否给未收集到的唯一属性值设置缺省值.如果"是", 单击缺省符号(Default symbol)图形框后,便可给图层对象更改符号.

6. 单击 Apply 按钮后,便完成这个 Value Map Renderer 的设置,当前图层属性已由前五步完成设置.

下面程序代码在 mo View 应用程序中,完成的工作是:从对话框中提取所有的设置并由此得到一个新的 Value Map Renderer.这段代码是 Layer Prop.frm 中的一个过程(procedure),有几个关键元素:grd Value Map 栅格控制器存放值图翻译器用的唯一值, Recordset (记录设置)针对的对象是 frm Main 表单中 Map Disp 图形控制器所指定的当前层. unique List 字符收集器收集由记录设置所确定区域中的所有唯一值.

用类别描述属性

若 Recordset 的 Field 有一连续数值就不一定能创建值图.典型地,唯一属性值的数目会非常接近甚至等于记录数目.

在一个类图(classifecation map)中,它将数值集合成一个个离散城,并且为每个城指定一个符号.下面是一个描述某地人口数量的例子. 假设要做一张地图,并且想用五种不同符号描述人口数的五个值城:0-9999 人用黄色画;10000-49999 人的用浅橙色画;50000-99999 人的用橙色画; 100000 人以上的用红色画.在这个实例中,用 3 个值 10000,50000 和 100000 划分为四城三类.用 Class Breaks Renderer object 便可创建这类图分类翻译对象(Class Breaks Renderer object)以下程序代码完成的工作是:给 Counties 图层创建并配置一个

Class Breaks Renderer object:

为生成并将 ClassBreaksrendere object 赋与名为 Counties 的图层，可用以下代码。

接下来便可使用 Class Breaks Renderer object 的其他属性.

一般说来,做一张类图是还要求给跨颜色的城着上颜色.使用 Ramp Colors 方法之前必须确定 Break Count (分类数)和 Break array (分类数组)的值. 而且既可用 Map Objects 颜色常数也可用 Visual Basic 颜色常数.作分类数

如果图层中有点或线的属性,那么使用 Size Symbols 方法可修改每个城对应符号的大小.这种方法完成的工作是:重新设置符号对象数组中各符号对象的大小记住,用 Size Symbols 方法之前必须先确定 Break Count 和 Break array 的值.

mo View 应用中 Class Breaks Renderer 的使用

mo View 应用中的 Layer property form 有一专用于 Class Breaks Renderer 的对话框. 下面是在 mo View 应用中创建一个类图的具体步骤:

1. 主菜单工具栏中单击 Map Contents 按钮.
2. 在图形工具栏中单击(Map Contents form)中增加一图层.
3. 在 Map Contents form 中单击 Properties...,然后,在出现的 Layer property form 中单击 Class Breaks 按钮.
4. 选择所感兴趣的值城. Mo View 应用程序扫描当前图层的所有区域然后列出所有值城. mo View 可能花上几秒钟来分类.
5. 在从红到黄的各种颜色中选择一种设置为缺省值. 如果想修改 Startcolor 和 End color,单击一图象框,出现专用于颜色的 Visual Basic 命令对话框.
6. 上述步骤完成当前图层的属性性设置.单击 Apply 按钮即完成了 ClassBreaks Renderer 的设置.

以下程序代码存于 Layer Prop.frm 中,完成的工作是:创建一个 Class Breaks Renderer 且在对话说明中设置其属性.

在这段程序代码中,一个 Statistics object (统计对象)在记录设置对象中被创建并产生了最小值和最大值.然后, 系统计算出等间距的一组城并将这组城提供给 Class Breaks Renderer 使用.

而且,提供了 Ramp Colors 和 Size Symbols 两种方法.注意, 对带有多边形特性的图层不能用 Size Symbols 方法.原因在于,当前图层中有多边形时,层属性表单隐藏了对 Size Symbols 的控制.

程序的末尾部分完成的工作是:用 cls Draw Symbol 类画 Grid control 内的符号.这个 cls Draw Symbol 类在得到一个符号对象和 Map control 之后先在 Map Control 上画该符号,然后在 Map Control 上用 out put Map 方法将这个符号再画到剪贴板(clipboard),最后粘贴剪贴板上的符号到 GridControl 的一个单元上.

用点密度模式画多边形

用多边形翻译属性一个有趣的方法是: 使用具有一定点密度值任意点模式.这种技术的一个应用是:创建一解说图的地图, 该图用不同级的区域反映人口密度.每个点位置本身并不能反映点本身有关属性,换言之,点的位置反映整个图上的布置以及一个多边形内的点平均密度, 这个多边形传递有关信息给图形取景器 (viewer).

Dot Density Renderer object (点密度翻译对象)

下面程序代码完成的工作是:为 Demo graphics 图层创建并设置一个 Dot Density Renderer object:

若 Draw Back ground 布尔值置为 True , 图层中的多边形应带有相应的当前符号和点模式.

Dot Value 属性即用一点来代表的区域的值,一个多边形区域中有 100000 个人,想要使用 200 点的随机模式,这时就可以使用 200 除 100000 或 500 除 100000 的点值.

Dot Size 属性反映屏幕上一个象素点的大小尺寸. Dot color 则反映点颜色.

Tag (标签)属性即描述应用要求的功能.

mo View 应用中 Dot Density Renderer 的使用

mo View 应用的 Layer property form 中有专用于 Dot Density Renderer 的对话框.下面是在 mo View 应用中创建一点密度图形的具体步骤:

1. 主菜单工具栏中单击 Map 控钮.
2. 在 Map Contents form 中增加一个有多边形特性的图层.
3 .在 Map Contents form 中单击 Properties...,出现 Layer property form 后单击 Dot Density.有一点要注意,如果该层不具有多边形特性, 就不会出现 Dot Density 这个选择项.
4. 选择所感兴趣的值域.
5. 选择点值,点大小和点颜色.看一下给点设值时的几点注意事项.点大小设置过小,造成的后果是要画的点增多得很多.一旦发生这种情况,按 ESC 退出画点的过程,但必须已经设定 Map Control Cancel Action 属性.如果要以多边形为背景,单击 Draw Paoy gons 检验盒至"开"状态.
6. 以上五个步骤完成翻译器属性设置.单击 Apply,设置过程完成.

用属性文本标签特性

渲染地物的另一种方法是用属性值描述文本.假设在一张地图上, 用点表示城市.除了给每个城市分配有一点符号外还写上城市名.而且,还应有一字段标明大小关系,即大城市用大号文本标签, 而小城市用小号且与大号文本不同的字型的文本来标签.

除了点,线,多边形属性外,还可用其它属性书写文本标签.Label Renderer 一个关键属性是 Symbol array (符号数组),其中的元素引用于 Text Symbolobject.Text Symbol object 中标明了用于翻译文本字符串的所有相关特性. The Label Renderer object

Label Renderer object 的属性

以下代码完成工作:创建并配置一 Label...给 Meters 图层:

如果 Draw Back ground 属性为 True ,图层属性连同标签正文一起显示.

使用 Label Renderer object,正文的有选择显示有更大灵活度.Height Field. Label Field, Rotation Field 及 Symbol Field 属性均为显示正文标签提供多个选择.如果图层中不出现这些可选择字段,可将它们设为临时 Tableobject 中的字段,用 Add Relate (添加联系方法)建立一个关系并计算部分或所有应用对话期间涉及到的属性.

Label Renderer 不能自动解决重叠文本标签之间的冲突. 缓和重叠文本问题的一个方法是使用小号字体,另一种方法是编写应用程序: 放大地理区域时便改变字体大小(变大).

Text Symbol object (文本符号对象)

Label Renderer object 用取自 Text Symbol object 的一个符号数组来写标签文本.如果给这个 Label Renderer 确定一个 Symbol Field,那么这个 Symbol Field 可用作这个符号数组的索引.如果没指定任何 Symbol Field, 那么所有的标签文本唯一确定为第一个符号,这个符号自成一数组,索引值为 0.

通过 Text Symbol object 中属性的设置,可产生两种文本:一种是依赖于规模的文本,即当放大文本时,文本大小增加;另一种是不依赖于规模的文本, 即文本在屏幕上保持文本大小固定不变.要得依赖规模的文本,应给 Height 属性设一非 0 值, Height 属性值用地图单位.要得到不依赖规模的文本,应给 Height 属性置给 0(缺省值),那么 Label Renderer 使用的是与 Text Symbol 有关字体.

mo View 应用程序中 Label Renderer 的使用

mo View 应用的 Layer property form 中有一专用于 Label Renderer 的对话框,下面是在 mo View 应用中用文本标签特性的具体步骤:

1. 单击主菜单工具栏中的 Map 按钮.
2. 在 Map Contents form 中增加一图层.
3. 单击 Map Contents form 中的 Properties... 按钮, 出现
   Layer property form 后单击 Labels 按钮.
4. 选择所感兴趣区域.
5. 在文本框中单击,便激活 Visual Basic 有关字体的常用对话框. 完成对话后文本框中出现新的字体.可更改的属性有: color,vertical alignments,horizontal alignment,rotation angle, 是否图层属性和图形符号来一起显示.
6. 上述步骤完成属性设置,单击 Apply 后 Label Renderer 设置完成.

图 4-20 mo View 应用实例中 Label Renderer object 的补充.

mo View 应用中并非所有属性都可操作. Height,Level Field, Rotation Field 以及 Symbol Field 都不出现在用于 Label Renderer 的对话框中.面程序代码选自 Layer Prop.frm 中 Apply Properties 过程,完成的工作是:创建和使用 Label Renderer object. 注意,如果不指定 Symbol Field, Label Renderer 用 Symbol array 中第一个元素来给所有标签正文产生 Text Symbol object.画形状和文本几何对象的用法之一是:在 Map Control 上

画形状和文本.

这个 Map Control 不属于 Geo Dataset .Draw Shape:获取一形状并用已确定符号画该形状.Draw Text:获取一文本串并用已确定 Text Symbol 沿着一点,一线或一个矩形来画该文本串.Track Grcle,Track Line,Track Polygon 以及 Track Rectangle 用来交互地得到供 Draw Shape,Draw Text 用的形状.

Draw Shape 法画以下几种几何对象: point(点),Line(线),Polygon(多边形),Rectangle(矩形)以及 Ellipse(椭圆). Draw Text 法画文本串有五种方式:在一个点上写文本串;沿一直线写文本串;在其他所有对象约束盒的中点上写文本串.

事先必须在以下四个 Map Control 事件中的一个事件中编写 Draw Shape 和 Draw Text.

下面提供几种途径传递完整的几何形状给 Draw Shape 和 Draw Text:

.第一种方法,在 Map Control 中使用 Track Circle,Track Line,Track Polygont 和 Track Rectangle 法完成形状的交互输入.

.第二种方法,程序化建立几何对象.例如, 可使用线或多边形点集的点添加法,先得到大致轮廓,再重画得到结果.要记住, 几何对象的所有坐标都在图形空间,所以使用 To Map Point 法便得到与控制坐标相关联的几何对象.

.第三种方法,画一个由 Recordset 产生的形状.为得到该形状对象,记住要使用 Recordset 中字段集合中 Shape 字段的值属性.

下面是一实例,说明如何使用 Draw Text 法并且沿着一线对象.

许多这样的程序代码教会你如何使用 Draw Text 和 Draw shape.在 mo View 实际应用中,可用图形工具栏来增加几何对象和简单文本.浏览一下 mo View 中 Edit Layer .cls 这个 Visual Basic 类, 文件夹中可看到两个项目: Draw Text 和 Edit.在 Map Objects 的联机 Help 中有使用 Draw Text 和 Draw Shape 方法的程序代码实例.

打印地图

有四种输出你在 Map Objects 中地图的方法.复印地图法将在地图控制器中显示的内容复印到剪贴板上.输出地图法将地图控制器中所显示的内容以一个位映象或提高的元文件格式写入一个文件.

输出地图法将地图控制器中所显示的内容提供给另外一个设备.比如一台打印机或另外一个窗口.打印地图法将地图控制器中所显示的内容提供给预设打印机.地图控制器中的地图输出方法.

察看叫作 Output (输出)的 Map object 样本设计以便了解所有这些地图控制器的方法是如何使用的.同时,察看 Send Map (传送地图) 样本设计来了解如何通过 MAP 将一幅地图以电子由件方式传输出去. MAPI, 是微软为电子邮件设计的应用编程接口的缩写.

Map objects 是面向用于在你的计算机屏幕上显示地图的,你可以将地图打印到任一个你已在 windows 中安装了驱动程序的打印机或绘图仪上.但 Map objects 并不善长于高质量的制图产品. 它缺少用来定义客线和填充符号的软设备也没有支持地图布置的内

在功能.如果你想制作发行级别的地图,请使用其它的 GIS 软件例如 ARC/INFO 或 Arc View

第五章

第五章　选择元素以及检索信息

地图是关于地理地点信息的存储库:它们的位置,空间关系以及属性值.

你可以通过学习如何察询与你的数集有关的表及通过掌握空间和逻辑的查询方法来从你的数据中获取信息.管理有关 Map object 的信息的关键就是记录集.它表示了你的地图层中的属性信息并向你提供了使用表记录的方法.

同时我们将学习如何查询记录集中的信息组,修改它们, 以及更新它们的值.并且我们将为了三种返回一个为选定地形核对位置的记录集的方法再次访问地图控制器.

在本章中,我们将涵盖以下主题:

使用记录集指导浏览与一个图层相关联的多个记录.

对一个记录集进行基本的数值统计计算.

查询及更改一个记录集中的信息组.

使用 ODBC 的项目表及关联.

在给定距离内搜索元素

使用从一个形状文件或 SDE 层中得到的形状来选择其它形状.

寻找重叠了地形的形状.

寻找多个在一条线或点相接触的形状.

寻找包含其它地形的形状.

使用一条选择语句来选择元素.

在前些章中我们已讨论了一些数据存取目标, 在本章中我们将集中讨论剩下的数据存取目标.

记录集是用来同一个图层中信息进行交互作用的关键目标, 一个记录集代表从一个地图层中得到的多个记录.

记录集有 Table Desc 目标和信息组集合两个特性,其中包括信息组目标.

从一个记录集目标中,你可以用基本的统计信息来计算一个统计目标.

表目标也可用来增加关联或进行地址匹配.

字符串集合包含有标准的字符串数据类型并且在更改用户接口控制和 Renderer 目标的一部分.

在本章的结尾,我们将看到应用在 mo View 样本应用程序上三条命令中记录集:Identify (辨别), Find (寻找)和 Spatial Search (空间搜索).

使用记录集

一个记录集目标是作为 Map Layer (地图层)目标的一个特性而被创建的,新的记录集也可以通过在 Map Layer (地图层)目标上运用方法而被创建.

有两种途径得到一个记录集目标.

当你创建了一个地图层目标时, 你就将一个记录集目标作为记录特性创建起来.这个记录集包括那个地图层的所有记录.

当你运用下述地图层方法之一时; Search By Distance (通过距离搜索)搜索表达式 (Search Expression),以及 Seafch Shape (搜索形状), 一个记录集将被该方法返回, 只有满足该方法条件的记录格被包含在该记录集中.

记录集是不可创建目标.也就是,你不能用 New key word (新关键字)或 Create Object (创建目标)功能来声明一个记录集.一个记录集仅能作为一个地图层的属性或通过应用三种方法之一而被创建.

记录集目标

这里提供了记录目标的属性及完整方法

当你建立了一个记录集时,其位置在第一个记录上.Move Next,Move First 及 Move Previous 法使你可在多个记录中浏览.EOF 特征返回你是否到达了记录集的结尾.

这里有一段简单的程序,它演示了一个记录集在一名叫人口普查的地图层上的基本用法. 这个例子将对此记录集中所有记录打印一称为家庭的信息组的值到 Visual Basic 下的 Debug 窗口.程序

注意 Visual Basic 也有一个运行情况相似称为记录集的目标.为了区别这此目标,用像这样的一个声明来充分限定你对 Map objects 记录集目标的声明将是十分重要的:

某些记录集可以被更改,其余则不然.察看可更改特性来确定你是否可以更改一个记录集中的记录.如果你希望更改一个与形状文件有关联的记录集,而可更改是舍己救人假的. 可以通过在你的文件系统中修改形状文件三位字节的只读状态如果你有特权这样做. 如果可更改是假的并且记录集与-- SDE 层有关联与你的 SDE 管理员商议以取得所需特权来更改 SDE 层.

如果一个记录集包含从一个关联表得到的信息组(通过在一 Map Layer 上运用 Add Relate 法)并且记录集可更改,你只可以更改从基本记录集中得来的信息组.你不能更改从关联表联接来的信息组.

统计目标

统计目标是通过一个记录集上的 Calcu late Statistics (计算统计) 法而被创建的. 其目的是给你提供一个关于一个记录集中一个数字信息组的基本的统计一览.

这里有一简单的程序片段,它从一个叫作人口普查的地图层中创建一个统计标并计算一个称为家庭的信息组的统计值.

注意被表示的统计数字在你创建统计目标的是有效的. 如果你更改记录集中的值, 你必须计算一个新的统计目标察询新的统计结果.

信息组集合和信息组目标

信息组集合和信息组目标表示了一个记录集中多列数据表格

正如同记录集一样, Misual Basic 同样也有一个信息组目标. 要确定像

如下来充分限定你对一个 Map Objects 信息组目标的声明:声明语句

这里有一简单的程序片段,它将显示有关-- 称为人口普查的地图层的所有信息组的名字及它们的类型代码.程序段

又如, 它设定一个信息组的值并将称为人口普查的地图层中由多边形表示的人口提升 5 个百分点.

在每个记录集中有一个称为"形状"的特殊的信息组.此信息组代表了地学数据集中地形的几何形状.如果此记录集是可更改的, 你可以对此信息组运用编辑方法并可立即在-- 形状文件内用另一形状来代替此形状. 这就是如何利用 Map Objects 进行简单的形

状文件编辑.信息组称为"形状" 的类型特性可是 mo Point,mo Line mo Poly Gon 类型常量的任一.

Table Desc 目标

你可以用信息组集合来察询信息组并且你可在记录集中修改信息组值, 是要修改一个信息给的特性,你就需要使用 Pable Desc 目标了.

Table Desc 目标向你提供关于与-- 记录集有关联的项目表中信息组的信息.它典型用于创建新--新记录集和形状文件.修改已存在的记录集的信息组定义,以及查询信息组特征.

举一个应用一个 Data Connection (数据联接)的 Add Geo Data Set (增加地学数据集,法同 Table Desc 目标的例子,请在 Map Objects 中察看为 Add Geo Data set (增加地学数据集)法编写的在线帮助范例程序.

表目标以及创建关键

一个表目标代表了从-- ODBC 数据源中得到的单个项目表

表目标是一个只读数据访问目标,它代表从-- ODBC 数据源得到个单个项目表.你可以将一个项目表作为一个关联加到 Map Layer 的目标上.

选择元素

在 Map Objects 中,你可以通过应用-- 地图层目标上的三种方法来选择元素.结果将以一个记录集的形式返回.

Search By Distance 和 Search Expression 很快就可解释完,但 Search Shape 法提供了如此丰富的一套搜索方法我们不久将要涵盖. 每种搜索方法的注解前均有其用法的句式.

形状自变量可以传递用鼠标在 Map 控制器上画的一个或许多个形状, 从 Map Layer 返回的形状.或被创建的有着程序计算坐标的几何目标.要注意的是从--Map Layer 得来的形状可以是在同一层或另外--Map Layer 中得来的.

通过距离来搜索元素

对在一个或多个形状的限定范围内元素进行定位.使用--Map Layer 目标上的 Search By Distance 法.距离值使用 Map Layer 的地图单位;英尺,米,或其它.Search By Distance 法接受一个或一集的点,线,矩形或多边形目标.并且选定所有全部或部分在限定距离内的元素.

你可以任意地用一 ANSI SQL 语句的哪里条款来进一步地限定选择果你不选择的话,那所有符合空间搜索的元素将在记录集中被选中.

通过询问方式搜索元素.

Search Expression 法接受一 ANSI SQL 语句的哪里项并且返回一包括所有满足表达式的元素的记录集.

通过形状来搜索元素.

Search Shape 法有三个自变量:

1. Shape 是一目标,它可以是一个点,一条线, 一个矩形或多边形目标或是这些目标的集合.这个目标可以从像 Mouse Down,Trackline,Track Polvgon 或 Track Rectangle 等地图

控制事件之一中定义,或通过勾画建设一几何目标,或通过重复通过一记录集并从"形状"信息组返回一个或一集合形状.

2. Search Method 是一整型量.为了你的方便,你可以输入搜索类型常数之一.如果你愿意,你可以输入搜索类型常量的整数值.

3. expression 是一字符串,它是一 ANSI SQL 语句的哪里条款.此表达式是任意的,但当你调用此方法时, 你必须放置两个引号来表示一空白表达式.

观察 mo View 应用程序中的 Spatial.frm 的程序,以看到在从地图控制器事件中的形状或从一地图层中得到的形状上应用 Search Shape 法的范例那程序的一部分将在本章末尾列出.现在我将研究 Search Shape 法的所有搜索类型.

形状和元素边界重叠

此搜索类型返回所有其矩形范围整个地或部分地重叠搜索元素范围的元素.此搜索方法的一个实际用法是在当前地图范围内选定所有元素.从一几何的通过 Map 控制器上的一鼠标事件构造的目标及从一个或一集从一 Map Layer 中得到的形状中为此种搜索类型构造搜索元素将是非常有用的.

形状与元素有共同点

此搜索类型返回与搜索元素至少有一个同一的共同点的元素.用此种方法同 Mouse Down 事件来选择几乎是不可能的.然而,当此种搜索方法应用在从一 Map Layer 中得来的点来给所有将那点的精确位置共享为一个多边形或线的顶点的其它元素定位时, 此搜索方法将是十分有效的.

形状和元素交叉边界

此搜索类型返回与搜索元素相交叉的元素.

从一通过地图控制器上的一鼠标事件构造的几何目标和一个或一集从 一 Map Layer 中得到的形状中为此种搜索类型构造搜索元素将是非常有用的.

形状和元素共享一公有线.

被此搜索类型返回的元素必须与搜索元素共享至少一条同一的公共线.

此搜索类型对从一个 Map Layer 中来的搜索形状是十分实用的,但对由地图控制器上的鼠标事件构造的搜索形状并不实用.

形状和元素共享公共点或交叉边界

此搜索类型返回与搜索元素(或多个搜索元素)共享一个共点, 或与之相交叉的元素.

从一个由地图控制器上一鼠标事件所构造的几何目标,和从一 Map Layer 来的一个或一集形状中为此搜索类型构造搜索元素将是十分有用的.

形状和元素相交叉

此搜索类型返回与搜索元素接触的,全部或部分地在搜索元素之内的,或全部或部分地包括搜索元素的元素.

从一个由地图控制器上由一鼠标事件所构造的几何目标和从一 Map Layer 中得来的一个或一集形状中为此搜索类型构造搜索元素是十分有用的

形状和元素在内部交叉

如果搜索元素是一多边形元素,此搜索类型返回全部或部分包含在其中的元素,但不是与之邻近的.另外,返回元素本身也应是多边形元素, 并且此法返回部分或全部地包括搜索元素的多个元素.

从一个由地图控制器上由一鼠标事件所构造的几何目标和从一 MapLayer 中得来的一个或多个形状中为此搜索类型构造搜索元素是十分有用的

形状和元素相交叉但没有相切边界

此搜索类型与 mo Ares Intersect 相同, 但搜索元素和元素的边界不可以交叉或接触.

从一个由地图控制器上由一鼠标事件所构造的几何目标和从一 Map Layer 中得来的一个或一集形状中为此搜索类型构造搜索元素是十分有用的

形状包含元素

此搜索类型返回完全包含搜索元素的元素.

如果此元素是一多边形元素,那么,搜索元素必须全部在其中,包括此元素的边界.如果此元素为一线元素,搜索元素必须沿着此元素的路线,如果此元素为一点元素.搜索元素必须在其一点顶点上.

从一个由地图控制器上由一鼠标事件所构造的几何目标以及从一地图层中得来的一个或一集形状中为此搜索类型构造搜索元素是十分有用的.

元素包含形状

此搜索类型返回完全包含在搜索元素中的多个元素.

从一个由地图控制器上由一鼠标事件所构造的几何目标以及从一地图层中得来的一个或一集形状中为此搜索类型构造搜索元素是十分有用的.

形状完全地包含元素

此搜索类型返回完全地包含在搜索元素的多个元素.不包括搜索元素的边界.此元素必须是一个多边形元素,搜索元素必须完全地在其中,并且它们的边界不可以交叉或相切.

从一个由地图控制器上由一鼠标事件所构造的几何目标以及从一地图层中得来的一个或一集形状中为此搜索类型构造搜索元素是十分有用的.

元素完全地包含形状

此搜索类型返回完全被搜索元素所包含的多个元素,并不包括搜索元素的边界.

搜索元素必须是一多边形元素,此元素必须完全在其中, 并且它们的边界不可以交叉或相切.

从一个由地图控制器上由一鼠标事件所构造的几何目标以及从一地图层中得来的一个或多个形状中为此搜索类型构造搜索元素是十分有用的.

元素包含形状的第一点.

此搜索类型返回包含搜索元素的第一个坐标位置的多个多边形元素.如果搜索元素是一条线,线的始点即被使用的位置.

如果搜索元素是一个多边形,那么始点与终点是相同的,并且被使用.

从一个由地图控制器上由一鼠标事件所构造的几何目标以及从一地图层中得来的一个或一集形状中为此搜索方法构造搜索元素是十分有用的.

形状包含元素的重心

此搜索类型返回其重心被形状所包含的多边形元素.重心是一多边形受重力的中心的位置.

从一个由地图控制器上由一鼠标事件所构造的几何目标以及从一地图层中得来的一个或一集形状中为此搜索类型构造搜索元素是十分有用的.

元素与形状是完全相同的

此搜索类型返回与搜索元素完全相同的元素."完全相同"的意思是指形状的每个顶点的坐标位置是完全相同的.

当判断元素是否完全相同时,此方法要考虑元素类型以及坐标描述. 也就是,一条与一多边形有着相同的点的集合的多边线与多边形并不认为其是等同的.此搜索类型典型用于寻找完全一样的数据.

仅从从一地图层中得到来的一个或一集形状中为此搜索类型构造搜索外形是十分有用的.

在 mo View 应用程序中应用记录集

我们现在将在 mo View 的范例应用程序中应用记录集以及它的支持目标. 你将看到一范例界面,从中你可以借鉴一些想法,同时还有在 Map Layer 目标上应用了我们已经讨论过的记录集目标和方法的范例程序.

在 mo View 中有三种形式应用记录集从一地图层中检索信息; Identify.frm,Find.Frm 和 Spatial.Frm.

mo View 应用程序中的辨别工具.

当辨别工具被激活时,你可以用光标拾起任何元素并检查它的信息组的值当你使用这个工具时,一个带有从选定元素中得来的属性值的表格将出现.如果你的搜索提供了不只一个极为接近的元素,那你将在一复合框中被提供对外形进行选择的机会.当你用毕辨别形式.你可以消除它.

注意在 mo View 中的主要工具条中一个按钮组中有五种工具: Zoom in Zoom out, Pan,Identify 和 Graphics.仅有这些键中一个可以被按下. 当你按下 Identify 时,当你在地图内敲击鼠标时元素就被选中了. 当辨别工具被激活时,地图控制器中的 Mouse Down 事件就启动辨别方式.

以下为辨别工具的程序遵循这此步骤来返回在一点选定的元素的有关信息:

1. 从 Mouse Down 事件接收 X 和 Y 位置作为过程变量.
2. 更新表格上的字幕,包括当前位置.
3. 为寻找元素设置一个 100 个地图单位的搜索公差.这是十分必要的, 因为如果没有一个搜索公差,寻找点和线元素将会是非常困难的.
4. 重复通过每个层来此点的搜索公差内寻找元素. Search By Distance 法被应用在一地图层上并且结果以一记录集形式返回.
5. 用从一记录集中选定元素中得到的信息组名和信息组值来填充一复合框.

下叙程序是从 mo View 样本应用程序中的 Identify.frm 中得来的.这些是那程序中一些关键控制和目标:

frm Identify 是 Identify.frm 的名字.控制地图器叫做 map Disp 并且在一称作 frm Main 的表格中.

CDO IDList 是有着所有所被找到的元素的表列的复合框.
Ist Feat List 为每一表列的一个信息组以及它们的值的表列.当你从
cbo IDList 复合框中选择另一个元素时此表列框框将更新.

mo View 应用程序中的寻找命令

mo View 中主工具条中的寻找命令允许你输入一个值. 然后它在你限定层中的所有信息进行搜索来进行字符串匹配.

当你敲击寻找钮来寻找含有一字符串为一信息组值的元素地时候.下叙寻找元素方式中程序遵循这些步骤:

1. 根据输入的字符串建立一搜索表达式.
2. 重复通过层集合中每一个层并且仅在选定层中搜索字符串.
3. 重复访问一层中每个信息组并且建立并应用搜索表达式.
4. 从被找到的元素中,用层名,信息组名和信息组值来填充表格控制器.

下面的程序是从 mo View 样本应用程序的 Find.frm 中得到的.这些是那程序中的一些关键控制和目标:

frmFind 是寻找表格的名字.

地图控制器被称为 map Disp 并且在一叫做 frm Main 的表格中.grd Feat List 是含有搜索结果的表格控制器.

mo View 中的空间搜索命令

当你敲击主工具条中的空间搜索命令时, 空间选择表允许你为选择元素限定一个层,搜索元素的类型,以及空间搜索方法之一.

当你应用一空间搜索时,下面的空间搜索方式中的程序遵循这此步骤:

1. 检查形状集合,如果是空的就退出.
2. 对限定的层运行搜索方法.
3. 用 After Tracking Layer Refresh 法中的 Draw Shape 法画出选定元素.

下面的程序出版自 mo View (人工操作视图)中 Spatial.frm (空间检索形式)的运用举例.这些是该程序中的一些关键控制器及目标:

frm Spatial 是空间检索形式的名称.

Map 控制器叫 map Disp ,置于一个名为 frm Main 的表格中.

g-search Shape 是检索形状的集合. g-selected Featwes 是选择特征属性的集合.

第六章

第六章　地址匹配及查找定位

在我们所建立和管理的文件中,地址及地点名称是最常见的地理信息数据库中可以根本没有诸如纬度值,经度值或图象坐标之类的空间信息,但却存有大量有关地理邮政方面的数据.计算机可根据地址确定的一些位置自动地建立一个绘图层,并能自动访问到大量地图中的属性信息.MapObjects 中包括一系列 OLEAutomation 目标,这些目标是专为地址匹配及根据地点名称查找定位任务而设计的.地址匹配是指在地图上找到并标明每条地址所对应的位置,它也称为地理编码.

下面是本章的概要:

. 街道文件的必备条件
. 交互式地址匹配

.地址转换及标准化
.批地址匹配
.查找十字路口
.查找所给地点名称位置并定位

AddressMatcher(地址匹配)目标要求你指定一个 GeoDataset(地理数据设置),其中包括街道中心线,地址变动范围以及分别适用于单个和成批地址匹配的方法的设定.AddressLocation(地址定位)目标中的代码具有表明一条地址是否被解析以及如何被解析的作用,如果地址已被解析,它还能显示匹配地址的地理位置.

匹配地址

AddressMatcher,Address 和 AddressLocation 目标用于交互式批地址匹配.首先,我们必须了解街道文件成立的具体要求,然后我们将讨论如何使用这些目标进行地址匹配.

用于地址匹配的专用文件

要进行地址匹配操作,你就必须有一个街道文件,其中包括具有地址信息的街道中心成路段.街道文件可以是一个图形文件.也可以是一个 SDE(存储分配部件)层.一个适用于地址匹配的街道文件必须满足以下三个基本条件:

1.街道文件中必须有街道中心线段,并在街道交叉处形成清晰的结点.除了街道中心线段外,街道文件中不能再有其它任何特征.

2.这些中心线段中的每一段都必须至少具备以下五个要素:(字段);街道名称,街道左边的起始地址,街道右边的起始地址,街道左边的结束地址,街道右边的结束地址.

3.右边的一对地址和左边的一对地址必须分别编上双号和单号.这也就是说,任何一边的起点和终点地址必须同时为双号或同时为单号;而每一对左右两边的地址编号则形成互补,即一单一双.显然,一条道路的左和右取决于你所行驶的方向,街道文件则遵循以下的协定:街道的左右视起点到终点的方向而定.

MapObjects 中的地址匹配目标是为美国使用而设计的.但只要你能够使用符合上述要求的街道文件,即使在世界其它地区,你也能够使用它们进行地址匹配.不过街道两边的地址编号必须是一边为双,一边为单.

另外,地址匹配目标也适用于中等长度的街道文件,如一个跨市中县的街道文件,但它们并不能设计成为大范围内的国家街道文件,比如一个囊括美国所有街道的文件.如果你的街道文件确实跨越了一个很大的地理区域,那么你可以自由使用一些地址中的城市名,邮政编码及乡村名以帮助你解决地址匹配问题.下面是街道文件睬图解一览表:

1.街道文件是由具有规定的左,右两边及起点,终点特征的中央线路组成.其中左,右边之分取决于由起点到终点的方向.

2.地址数据是按各个路段储存起来的.储存的 4 个数值有:Left-From(左起点),Right-Trom(右起点),Left-To(左终点),Right-To(右终点).在美国,街道一边的地址号为双数,而另一边则为单数.

3.地址匹配,或地理编码,就是一个通过地址中某路段的起始,终了位置,并同时考虑到单双号因素,以确定地理位置的过程.

4.如果你指定了一个分支,那么估算出的地址位置将落在该路段垂直线上的左边或右边.

5.除了返回一个位置,地址匹配还返回标准化地址数据值,包括房号,街道名,首字符串和尾字符串,房号和邮区号码.

在美国,你可以使用由 TIGER(地理地形综合密码参考系统)文件衔变而来的图形文件,这种文件由人口普查署以低廉的价格提供,或者你也可以使用由销售商提供的其它一些文件,这些文件有的是由 TIGER 文件增强功能后的新版本,有的则是完全由他们自己新开发出的版本.通常使用由商业渠道提供的街道文件要比你自己再去建立一个要快捷有效得多.

在应用过程中,你也许不会有一些其它包括附加通路信息的图层,如街道镶边石线这类信息,但

这些必须与用于地址匹配的街道文件分离开来成为一个独立的图层.实际上,你会有一个可见的绘有镶边石线的图层以及一个不可见的绘有街道中心线的地址匹配图层.

街道绘制文件

当你利用街道文件和地址匹配工作时,如果你使描绘道路镶边石的详细的设计图层可见,而使用于地址匹配的街道中心线图层不可见,那么你就能绘制出一个精彩的界面.然而,那些绘有详细的街道边饰的设计图层通常并不很实用.这里有一个帮助你在实际运用过程中使绘制出的街道文件实现的技巧.

表格左边的街道文件是用一根简单的 1 单位宽度的细实线绘制的.而表格右边,同样的街道文件被绘制了两次.首先用 5 单位宽的粗黑实线绘制一次,然后再用 3 单位宽的白实线绘制一次.将两个图层重叠起来就形成了双线道路的效果

只要用几分钟的时间,你就可以在 MapContralProperties(图象控制功能)窗口自己绘制出这张图.绘制两张同一街道文件的图层,如上所述分别使用不同颜色及粗细的线条,然后叠加起来.将黑线的图层放在下面.

如果你用这种方法绘图,建议你最好仅在某一区域内使用双线,以免交叉引起混乱.在更小的范围内(当你拉近时),用单线绘制即可.通过在图象控制器的 BeforeLayerDraw(前图层绘制)操作中编写代码来估算当前图形范围,定义符号目标(这些符号目标与街道文件的 MapLayer(图层)目标相关联)的可见性及大小尺寸,就可以有条理地绘制出一个街道文件.

如果使用 ValueMapRenderer(图形赋值)将道路分类绘制,如公路用较粗的线绘制等等,那么你就能更有效地利用这些代码.这样绘制街道文件时,一定要注意规定好不同线条各自的比例尺寸.

地址匹配目标

在 MapObjects 中,AddressMatcher 目标控制整个地址匹配过程.

1.BuildIndex(建立索引)为 StreetTable(街道表)建立一个地理编码索引.FindAllStreetNames(查找所有街道名)查找指定字符串开头的街道 FindApproximateMatches(查找近似配对)返回与指定街道相似的街道名.GeoCodeTable(地理代码表)用于你所指定的表中的批地址处理.

2.ExactMatches(精确配对)指明 MatchAddress(匹配地址)方式是否使用近似匹配.Indexed(检索)返回 StreetTable 是否已有地理编码索引.Offset(距离)指明与街道中心线段的垂直距离.

3.MatchAddress 方法将 StreetTable 中指定的地址配对并通过 AddressLacation(地址定位)将结果返回.MatchIntersection(交叉点匹配)方法查找两条街道的交叉点.StandarolizeAddress(地址标准化)方法解析地址字符串并通过 Address 目标将标准化的地址返回.

4.StreetTable 是街道网络的 GeoDataset.Valid(有效的)表明 StreetTable 和所需字段名是否有效,可以进行 MatchAddress 和 MatchIntersection.这些功能指定了 StreetTable 的字段名以进行地址匹配 CityField(城市字段).CountryField(乡村字段),House-CoordinataField(房屋坐标字段),HouseSuffixField(房号尾字母缩写字段),LeftFromField(左起点字段),LeftToField(左终点字段),LeftzipField(左邮政区字段),PrefixDirection-Field(方向首字母缩写字段),RightFromField(右起点字段)RightToField(右终点字段),RightZipField(右邮政区字段),StateField(国家字段),StreetField(街道字段),StreetType-Field(街道类型字段),SuffixDirectionField(方向尾字母缩写字段).

利用它,你可以找到以指定字符开头的街道名返回与指定街道名相似的街道名,运用批地址匹配和交互地址匹配,找到两条街道交叉的地方,并使地址标准化.

下面是使用 AddressMatcher 目标的一般步骤:

1.指定一个 StreetTable 作为 GeoDataset 目标(该目标是通过 DataConnection(数据连接)目标

建立的).StreetTable 必须满足前文所述街道文件的要求.

2.指定出你的街道文件中出现的地址内容,范围的部分或全部加以定义.你至少需要定义这几个字段:LeftFromField,LeftTo-Field,RightfromField,RightToField,和 StreetField.

3.给 StreetTable 建立一个索引.利用 Indexed 功能核实一下这个索引是否已经存在.

4.运用下述方法中的一个进行一下你想要实现的地址或街道匹配类型的操作:FindAllStreetNames,FindApproximateMatches,GeocodeTable,MatchAddress,MatchIntersection 和 Standardize-Address.

AddressMatcher 的有些功能有缺省值,如果它们符合你街道文件中的字段,那么不必去改变它们.这些缺省值是:

LeftFromField 的缺省值为 L-F-ADD
LeftToField 的缺省值为 L-T-ADD
LeftZipField 的缺省值为 ZIPL
PrefixDirectionField 的缺省值为 PREFIX
RightFromField 的缺省值为 R-F-ADD
RightToField 的缺省值为 R-T-ADD
RightZipField 的缺省值为 ZIPR
StreetField 的缺省值为 NAME
StreetTypeField 的缺省值为 TYPE
SuffixDirectionField 的缺省值为 SUFFIX

交互地址匹配

下面是 MapObjects 中进行交互地址匹配的步骤:

为 AddressMatcher 目标指定一个街道网络作为 StreetTable 的属性.这是一个 GeoDataset,包括街道中心线.这些街道都有一组标准化字段提供街道及地址信息.

2.检查 Indexed 属性以确认 StreetTable 的地理编码索引是否已建立.如果没有,则利用 BuildIndex 方法建立一个.

3.检查 Valid 属性以确认 StreetTable 和指定地址字段(未显示在这张图形中)是否符合地址匹配的要求.

4.进行交互地址匹配最简单的途径是通过 MatchAddress 方法直接用一个地址字符串.AddressLocation 目标被返回.如果成功了,你可以从 Location 属性中找到地理位置.反之,你可以从 Matchcode(匹配代码)中找到失败的原因.

5.进行交互地址匹配的另一条途径是利用 StandardizeAddress 方法首先建立一个 Address 目标,然后在 Address 目标的基础上使用 MatchAddress.利用这种方法,你可以访问到邮政编码,城市以及标准化的房号和街道名之类的地址信息.一旦你的地址匹配成功,你就能够进行 Map 控制器中的绘图操作,如用 AfterTrackingLayerDraw(跟踪后图层绘制)在地址匹配成功的位置画一点.

这里有一个简单的代码片断,是在一个名为"C:\Files"的文件夹中查找"Streets"这个图形文件,并在一个固定版本字符串上进行匹配.

注意在这个分段码中无任何可参考的图层.事实上,不使用 Map 控制器你也能够实现地址匹配,只要在 MapObjects 中进行 OLEAutomation 目标操作即可.另外,在这个分段码中几手没有什么判断和检验的过程.与其麻烦地为图形文件名和文件夹编号,还不如操作 VisualBasic(可视基础)共用对话来指定文件.

地址定位目标

当你在 AddressMatcher 上应用 MatchAddress 方法的时候,一个 AddressLocation 目标也就建立

了.如果匹配成功,内插位置可作为 Point(点)从 Location 属性被返回.查阅 Matched(已匹配的)属性以确定匹配是否已查找到.

1.Location 是已匹配地址的地理位置.Matched 表明地址匹配是否成功

2.MatchCode(匹配代码)返回这些常数中的一个(整型),分别说明地址是如何被解析的或是地址匹配失败的原因.

3.ExactMatches 的状态决定地址匹配是否能被调出修改错误.Offset 指定个垂直位移值用来估计一个街道左边或右边的位置.

4.StreetSide(街道两侧)以整型的形式返回这些常数中的一个,表明地址是在街道的哪一侧被发现的.

StreetSide 属性返回常数值 moLeftSide 或 moRightSide.该值表明地址是在路段的左边或是右边,左,右的从起点到终点的方向为准.

评估地址匹配

如果你设定 AddressMatcher 目标的 ExactMatches 属性为"真",那么街道名的拼写必须是正确的.常数 moMatchSuccess 会被返回(表明成功),或是以 moMatchFailed 开头的常数被返回(表明失败)且说明失败原因.

如果你设定 ExactMatches 为"假",那么寻找匹配的要求就不如此严格,MatchAddress 方法会利用逻辑来辨别可能的正确拼写从而获得地址.有 7 个以 moMatchSuccess 开头的常数返回时表明成功状态,其后的代码则表明地址是如何被解析的.

下面是可被 AddressLocation 返回的所有表明 MatchCode 属性的常数

moMatchSuccess 表明一个确定的地址匹配已查找到.如果 ExactMatch 设定为"真",那么这是唯一可被返回表明成功的常数.

moMatchSuccessUnresolredAmbiguity 表明地址已配对,其中一条配对地址被 AddressLocation 返回,但还有其它候选匹配.例如,如果你键入一条地址"1500Maple",这个常数就会被返回,因为街道文件中既有"MapleDrive"又有"MapleCourt",而 1500 对于两条街道均是有效房号.然而你无法断定到底哪个候选项被放置在 AddressLacttion 目标中了.

moMatchSuccessResolredInsertionError 表明地址被删去一个多余的字符后被配对.例如,输入的是"318S.MichiiganSt",然而"318S.MichiganSt"才是正确的地址.

moMatchSuccessResolredDeletionError 表明地址中插入了一个字符后被配对.例如,"1516Mantcello"被输入,然而正确的地址应该是"1516Manticello".

moMatchSuccessResolredSubstitutionError 表明地址中有一个字符被替换后才配对成功.例如,输入的是"1366PaygeLn",然而"1366PaigeLn"才是正确的.

moMatchSuccessResolredTranspositionError 表明地址中有两个字字符被交换位置后才配对成功.例如,"1224MiarMante"被输入,然而"1224MiarMante"才是正确的地址.

moMatchSuccessResolredPhoneticError 表明地址是在被找到了一个语音替换后才匹配成功的.例如,输入的是"816EastHiStreet",然而正确的地址应该是"816EastHiStreet".

moMatchFailedStreetNotFound 表明没有找到指定名字的街道.如果 ExactMatches 为"假",则表明计算机已试图分析街道名错误拼写的可能原因,但仍然是失败了.

moMatchFailedAddressRangeMismatch 表明指定街道名中所有路段的有效范围内都找不到要匹配的房号.例如,输入地址为"428WestFern",然而 WestFern 中的地址到 280 号就结束了.

moMatchFailedZipMismatch 表明街道表的邮政编码字段中没有地址中指定的邮政编码.

moMatchFailedComparisamFieldMismatch 表明相似字段,如街道的后半部分字符,是无效地址.例如,输入"123MainSt",然而街道文件中只有"MainAve"却没有"MainSt".缺省的相似字段有 StrsstTypeField,PrefixDiectionField 和 Suffix-DirectionField.你也可以指定其它字段来作比

较.如 City-Field,只要你从街道文件中选定一个字段名即可

moMatchFailedVnresolredAmbiguity 表明有多个匹配选项.这个代码与 moMatchSuccessVnresolredAmbiguity 意思一样,只是因为 ExactMatches 属性设定为"真",所以匹配失败,该常数被返回.

moMatchFailedUnknownReason 表明匹配失败,但原因不明.你交发现 MatchAddress 方法会帮你改正出地址中的大部分拼写错误,但不是全部.这一例外是,MatchAddress 不改动街道名的第一个字母,也不会同时显示上面所列 MatchCode 常数中两个以上的代码.但有一种特殊情况例外,MatchAddress 用同音代换改正街道名时也会改动第一个字母,如将"Phern"改成"Fern".

地址目标.

MapObjects 可接复你输入任何一个地址(正确的)并返回标准化后的地址.

这表明地址中所见包含的各项,诸如房号,街道首字符,街道名,街道类型,方向字尾等等,都已分类且标准化.例如,街道类型."Arenue"可被输入为"AV","Aer",或"Avenue",而街道类型将被标准化为"AVE".AddressMatcher 目标中的一种方法是 StandardizeAddress.这种
方法返回的 Address 目标中的各项均已标准化.

之所以要使地址标准化就是为了便于核实并找出潜在的错误及模棱两可不清楚的地方.另外,你还能利用标准化地址对数据库同类型的数据进行定位和修正.

标准化地址

下面就是 StandardizeAddress 方法如何处理地址字符串的:

上示图解中并未列出 Address 目标中的所有字段.其它标准化字段还有 HouseCoordinate 和 HouseSuffix.

在我们讨论交互地址匹配时,你已经看到了一些如何修改街道名中拼写错误
的例子.下面这一小段代码显示出修正后的街道名是如何被返回的.

当地址目标 CurAdd 由 StandardizeAddress 方式指定时,它的 Street 属性中有一值 Kearns.

假定当前街道文件中没有 KearnsRoad,然而 KernsRoad 是个有效街道.你也许认为将地址字符串或 Addres 目标输给 MatchAddress 方式都可以.但在这里,我们应该将 CurAdd 这个地址目标输给 MatchAddress 方式.接下来,地址目标中的 Street 字段值就将被修正为 Kerns.注意,如果你想得到一个正确地址的话,就必须输入一条地址目标而不是地址字符串.

这就是获得修正地址的方式.当--以地址匹配成功后,你可以利用地址目标中的结果更新数据库中的地址数据.

批地址匹配

MatchAddress 方式适用于单个地址匹配,而对于一次进行大量地址匹配就不那么得心应手.对于批地址匹配,一般使用 Address-Matcher 目标下的 GeoCodeTable(地理代码表),方法如下:

1 建立一个新的 DataConnection 目标,将其 Database 属性设置在将有图形文件或 SDEDatabase 的文件类中.

2 建立一个 AddressMatcher 目标,将其 StreetTable 属性设置在带有地址信息的 GeoDataset 中.

3 检查 Indexed 属性以确定街道文件是否已建立索引.如果没有,则用 BuildIndex 建立一个.

4 建立一个 Table 目标,并将 Databose 和 Name 设置到带有你需要匹配的地址的一个 ODBC 数据源中.

5 利用 GeoCodeTable 方式连同表的名字,该表包括地址以及一个包含需要匹配地址的字段.这种方式将利用你所指定的 DataConnection 目标列出一张新表,它的名字由你指定.被列为 StreetFields(街道字段)的字符串集所指定的字段是你想要从 addressTable(地址目录)复制到

outputTable(产品目录)中去的字段.

当你用 GeoCodeTable 进行地址匹配时,你是否需要一个有 MapLayers(图层)的 Map 控制器是任意的.你可以就用 OLEAutomation 目标(如前文所述)进行地址匹配操作而无需什么可见的图层.

在 on-linehelp(行上提示)或 MapObjectsProgrammer'sReference(MapObjects 操作指南)中查征 GeoCodeTable 的一个例子 VisualBasic 代码,看看这此步骤是如何以代码形式实现的.

查找道路交叉点

利用 MatchIntersection 方式查找两条街道的交叉点.

1 建立一个新的 DataConnection 目标,并将其 Database 属性设置在一个带有图形文件或 SDEDatabase 的文件类中.

2 建立一个 AddressMatcher 目标,将其 StreetTable 属性设置在带有地址信息的 GeoDataset 中.

3 检查 Indexed 属性以确定街道文件是否有索引.如果没有,则用 BuildIndex 建立一个.

4 利用 MatchIntersection 进行查找,同时要输入两个你需要确定交叉点的街道名称的字符串.

5 返回 Paint 目标.如果其值为 Nathing,则表明未找到交叉点,否则 X 和 Y 值就是街道交叉点的位置.

如果查找成功,MatchIntersection 方式就会返回一个 Paint 目标.实际应用时,你可以在 Map 控制器下利用 DrawShape 或 FlashShape 方式在你的图面上将该点画上.

使用地点名称

不用其他地址匹配目标而独立使用 PlaceLocater 目标也可以返回地点名称所在的位置.你可以单独为 PlaceLocater 目标建立一个街道文件,但实际上只要是含有地点名称字段的 GeoDataset 它都能接受.

你可以利用 PlaceLocater 目标建立的这种工具有时也叫做地名一览表.用一个引导类型或批选一个地名,光标就会自动移到指定的位置.

地点定位目标

下面是如何使用 PlaceLocater 目标的一览表:

以下是使用 Placelocater 目标的一般步骤:

1 为 PlaceLocater 目标的 PlaceNameTable 属性指定个 GeoDataset.这个 GeoDataset 只要包含有地名字段即可.

2 利用 BuildIndex 方式为地名建立索引.查看 Indexde 属性以确定该索引是否已经建立.

3 利用 PlaceLocater 目标中的一种方式.

利用 FindAllPlaceNames 返回所有以指定子字符串开头的地名字符串集

利用 FindApproximateMatches 返回与指定地名相似的地名字符串集.

利用 Location 返回是有指定地名特征的所有位置的点集.

使用 GeoDatasets 时,点,线和多边形都可作为 PlaceLocater 的 Place-NameTable 的属性.如果 GeoDataset 采用线的形式,则 Locate 方式返回的位置都是沿线的起点.如果 GeoDataset 采用多边形的形式,则 Locate 方式返回的位置是多边形的自己中心位置.

第七章

第七章　用 VisualC++来工作

本章供使用 MicrosoftC++4.0 版本编写 C++应用程序的程序员阅读.

VisualC++开发环境也可称为 Microsoft Developer Stdio;在本章中,我们将开发环境指定为 Developer Studio;我们将假定你对 VisualC++和微软基本类库(MFC)都较熟悉.如果你还没使用过 Developer studio,或者你在另一个环境中开发,通过阅读本章你也可得到一些启发.如果你有 Developer studio,但是以前没有机会用它来工作,那么在阅读本章之前你也
许要学习一下 Microsoft'sscribble Application 的教程,Scribble 教程介绍了一些重要的类库概念和使用方法,并且说明了如何在 Developerstudio 中使用 wizards 和资源编译器.可以在网上或者在随同 Developerstudio 的微软书籍<<Visual C++ Tutorials>>找到这个教程.

本书前面的章节告诉你在 Visual Basic 开发环境中如何建立使用 MapObjects 的应用程序.在本章中,通过一个创建一个可移可放缩的简单例子,我们将谈到在 Developer Studio 中使 Map Control 的几个话题.在 Samples 中的 Usa map 里可以找到一个涉及面更广的 C++例子.

这些是我们讨论的话题:

.使用 AppWizard 在对话中建立一个并入 Map Control 的工程.

.使用 Component Gallery 把 MapObjects 插入你的工程.

.把 MapObjects 提供给 C++ Wrapers.

.使用 ClassWizard 加入通过工程事件处理程序使用 MapObjects 的代码

应用程序概观

当你完成了初始的程序设计,一般你要执行下列任务来用 DeveloperStudio 和 MFC 库开发一个应用程序.

你使用 AppWizard 来创建一组初始文件和相关的 Windows 资源.用来建立用户界面对象的资源编辑器,如菜单和对话,以及用来产生编辑应用程序专用代码的 DeveloperStudio 的元素.

我们在本章中要讲的拜访 MapObjects 控制的方法属性的基本步骤是:.创建 OLE 控制载体工程.

.将 MapObjects 控制插入 OLE 控制载体.

.定义成员变量.

.用 urapper 类的预定义成员函数设计 MapObjects 的 Map 控制.

你也许已有一个应用程序,你想加入 map 成份,在本章中我们只关心 MapObjects 的 Map 控制,我们将不会谈及怎样处理与你的应用程序其它方面的交互作用.

创建一个 OLE 控制载体的程序.

这一部分告诉你如何使用 AppWizard 来创建一个基于对话的可执行工程,可用它来支持 MapObjectsOLEControl 的插入.我们把此程序称为 SuperMap.

为 SuperMap 应用程序创建初始文件.

1 在 File 菜单中,选择 New.New 的对话框出现了.

2 选择 ProjectWorkspace 然后单击 OK.NewProjectWorkspace 对话框出现了.

3 在 Name 框中,键入 SuperMap.AppWizard 用这个名字在 Location 一框指定的主目录下创建工程的目录.工作空间配置文件和工程的程序创建文件也都用该名.即分别名为 SuperMap.mdp 和 SuperMap.mak.

4 在 Platform 框中除 win32 还有其它检验盒的话,清除它们.

5 在 Type 列表框中,确定你选的是 MFCAppWizard.(exe).

6 单击 Create.

AppWizard 创建工程目录,MFCAppWizard-Step1 对话框出现.

7 在 Step1 中,单击合适的键来指定你想创建一个基于对话的应用程序;然后击 Next.

8 在 Step2 中,选择 OLEcontrols 检验盒;然后单击 Finish.NiwProjectInformation 对话框出现,它汇总了创建工程时 AppWizard 为你产生的设置和特征.

9 在 NewProjectInformation 对话框中单击 OK.AppWizard 创建了所有的文件并打开了工程.

观察工程

你可以通过在 ClassView,FileView 和 ResturceView 之间切换看到工程的不同方面.在 ClassView 里可以看到 AppWizard 创建的 SrperMap 类和成员的图示.在 FileView 中,可以看到 AppWizard 生成的工程 SuperMap 的文件的列表.ResowceView 列出为 SuperMap 生成的资源.

往你的应用程序加一张地图

在本章第一部分你为应用程序创建了框架.它包含 AppWizard 为你创建的缺省对话框.它是(SuperMapDlg 的一部分).在这一部分,我们将修改对话框.切换到 ResourceView--你将为你的应用程序加一个 map.

修改对话

1 在 ResourceView 中,打开 Dialog 文件夹列出资源.

2 双击名为 IDD--SUPERMAP--DIALOG 的资源.AppWizard 创建的系统设置的对话出现了.它有一个 OK 按钮.一个 Coneel 按钮,和一个静态文本区.

3 选定表上的所有控制然后把它们删除.

4 把对话框变为大约原来大小的二倍.

把 MapObjects 的 Map 控制加入 OLE 控制载体工程.

在你能从 OLE 控制载体拜访 MapObjectsControl 之前,你必须用 ComponentGalleny 把它加入工程.为此,使用 ComponentGallery 的主对话框的 Insert 按钮.Insert 把相关的头文件(.H)和工具文件(.CPP)加到当前被选的缺省工程中并更新 ProjectWorkspace 窗口中的信息.但,DeveloperStudio 以后不能这样处理所有的 MapObjects 的类型.我们得用我们提供的文件代替 Wizard 产生的头文件和工具文件.这个过程将在本章后面的章节"且取代 Wizard 产生的头文件和工具文件"中被介绍.

把 MapObjects 的 Map 控制插入你的工程

1 从 Insert 菜单中,选择 Component.ComponentGallery 出现了.

2 如果有一个以上的小窗口,用鼠标或者 CTRL+PAGEUP 和 CTRL+PAGEDOW 键选择 OLEControls.

3 用鼠标或者上下左右键移到 MapObjectsMapControl.

4 选择 Insert 按钮.

ConfirmClasses 对话框出现.这个对话框列出了为这个控制生成的类.按如下更改头文件和工具文件类的名称.

5 选择 CMap1 把它的类名改为 MoMap.

6 不选择其它的类.

7 把头文件名 Map.h 改为 mapObjects.h.

8 把工具文件名 Map.cpp 改为 mapObjects.cpp.ConfirmClasses 的对话框的修改.

9 单击 OK 然后单击 ComponentGallery 的 Close.代表已按装的控制的图标出现在 DialogEditorControls 的工具栏中

当你做完了这个过程,ComponentGallery 生成的被称为 wrapper 类的一种类被加入你的工程.你的应用程序使用 MoMap 这个类作为控制载体 supermap 和嵌入的 MapObjects 的 Map 控制之间的界面.

当你已把 MapObjectsmap 控制插入你的工程后,把控制的一个实例插入到应用程序的对话框中.

把 Map 控制加入对话样板中.

1 如果 ResourceView 不是当前视.在 ProjectWorkspace 窗口中单击它.

2 打开 Dialog 文件夹.

3 双击主对话框样板.IDD-SUPERMAP-DIALOG.

4 选择 MapObjectsMapControl 图标(在 Controls 工具栏中).

5 在对话框里面单击一下把 MapControl 插入,把它调到覆盖对话的 3/4.

6 在 File 菜单中,选 Saveall 以保存所有对话框样板的修改

为 Map 设置一此初始的属性.

1 右击 Map 在出现的菜单中,选择 Properties...MapObjectsMapControlObject.

2 在 MapObjectsMapControlProperties 对话框中,单击 Control 标志.

3 单击 Add 并把 Samples\Data\World 文件夹置于配置媒体或盘中.

4 双击 Country.shp 把它加到 Layers 集合中.

5 在对话框中选择 country 并单击 properties.LayerProperties 对话框出现.

6 选择一种颜色再单击 OK.

7 单击 All 标志把 Appearance 属性的值改为 1-3D.注意在发行后这个属性在 Windows95 上是可工作的,在 WindowsNT 上不行.

8 关闭对话框.

9 在 Build 菜单中,选择 BuidAll.建立完成后,按 F5 或单击工具栏中的 GO 按钮来看对话框中的 Map.有 MapObjectsMapControl 的 SuperMap 对话框.

祝贺你!你的应用程序里有一张图;然后,这还不算什么--你肯定还想加入其它的能力.为此,关闭 ProjectWorkspace.

工程的修改

为了使应用程序能够拜访 Map 控制,ComponentGallery 自动的把 wrapper 类.cpp 文件加进工程,把 wrapper 类.H 文件加进对话框工具文件

为得到和设置属性(和调用方法),wrapper 类提供了所有方法和属性的声明.在发行时,ClassWizard 不能为一些 MapObjects 目标创建 wrappers,由此我们提供了包含 wraper 类完整工具的 MapObjects.cpp 和 MapObjects.h.你将需要用这些文件代替 Wizard 产生的文件.它们位于 \Samples\CPPWrappers 文件夹中.

取代 Wizard 产生的头文件和工具文件.

1 确定你已将 ProjectWorksPace 存盘.

2 关闭 DeveloperStudio.

3 用\Samples\CPPWrappers 中的名称相似的文件取代 ProjectWorkspace 中 MapObjects.CPP 和 MapObjects.h 的版本.

4 重开 DeveloperStudio.

5 如果你在 SuperMapClasses 没有看见所有的 MapObjects 目标,关闭 DeveloperStudio 并删除工程的.ncb 文件.这个文件包含了由分析程序产生被 ClassView 和 ComponentGallery 使用的信息.DeveloperStudio 在你重开工程时将自动的重新产生这个文件.

ClassView 将显示所有 MapObjects 目标.

Wrpper 类头文件

Wrpper 类头文件定义了 MapObjects 的 Map 控制的敞开界面.这里是头文件的一个引述,表明了

MoRetangle 类的敞开界面.

这些函数可以使用一般 C++语法的其它程序中调出.

在工程中加入一个成员变量

既然你已把 Map 控制加入工程并把它嵌入一个对话框载体里,工程的其它部分就能够拜访它.拜访这个控制最简单的方法就是创建一个对话类的成员变量,CSuperMapDlg,和被 ComponentGallery 加入工程的 wrapper 类类型相同.以后你就可以随时用成员变量来拜访这个控制了.

往对话类加一个成员变量

1 装入 SuperMap 工程.

2 从 menu(菜单)视中,选择 ClassWizard.

3 选择 MemberVariable 标志.

4 从 ClassName 下拉式列框中,选择主对话框类,CSuperMapDlg.

5 从 ControlIDs 下拉式列框中,选择嵌入 Map 控制的控制 ID,IDC-MAP1

6 单击 AddVariable 按钮.AddMemberVariable 对话框出现.

7 在 MemberVariable 编辑框中输入一个名字.例如,用 m-map.

8 在 Category 下拉式列框中,选择 Control.Variabletype 编辑框中自动的写入控制 wrapper 类名字.

9 单击 OK 关闭 AddMemberVariable 对话框.

10 单击 OK 接受你的选择并退出 ClassWizard.

成员变量给工程带来的修改

当 ClassWizard 把 m-map 成员变量加进工程,它也给 CSuperMapDlg

类的头文件加入下列几行:

而且,ClassWizard 把一个 DDX--Control 调用加进 CContainerDlg 的 DoDatoexchange 工具.

设计 OLE 控制

现在,你在对话样板插入 Map 控制并为它创建了一个成员变量.你现在可以用一般 C++语法来拜访嵌入控制的方法和属性.前面已经说到,wrappwr 类头文件包括一系列成员函数,你可用它来得到和设置任何属性值.而且,头文件包括了方法的成员函数和所有定义 MapObjects 有名常数的 enumeration 这里是从头文件的 Enumeration 部分的一个引述.

一般在主对话框类的 OnIntiDialog 成员函数里修改控制属性.这个函数在对话框出现之前被调用并用来初始化它的内容,包括任何控制.

下面黑体形式的代码用 m-map 成员变量来修改嵌入 MapControl 的 MousePointer 的属性,按所示插入黑体代码.

创建处理例行程序

除了设置在 OnIntiDialog 成员函数的 Map 属性外,你还可以为在 SuperMap 对话框的每个 ObjectIds 包括 Map 而其它控制创建处理例行程序(成员函数),你可以使用 ClassWizard 的 MessageMaps 标志或者 WizardBar.我们先用 ClassWizard 给 Map 加一些移动和放缩的功能.

创建一个处理例行程序

1 从 View 菜单中,选择 ClassWizard.ClassWizard 出现了,它显示了当前所选类或你上项编辑的类的信息.

2 选择 MessageMaps 标志.

3 在 ClassName 下拉式列框里,选择 CSuperMapDlg,你将工作于其上的用户界面成份的类名.

ClassWizard 显示了对话框的信息.

4 在 ObjectIDs 框中,选择用户界面目标的名称,这个目标是你想为之定义一个信息处理程序的目标.我们正用 map 工作,所以选择 IDC-MAP1.

5 在 Messages 框中,选择 MtuseDown 作为你想为之定义处理程序的信息.选择 AddFunction(或双击信息名)一个已定义处理程序的信息以粗体显示出来.

6 如果 AddMemberFunction 对话框出现,键入 OmMapMtuseDown 作为成员函数的名称然后打回车.

你将返回 ClassWizard 的 MessageMaps 标记.ClassWizard 用黑体显示信息名表示它已定义了一个信息.新的信息处理程序名出现在 MemberFunction 框中.

7 选择 EditCode 跳到 ClassWizard 刚产生的空函数体中并开始定义函数的行为.当你选择了 EditCode,ClassWizard 按如下更新源代码.它在头文件插入一个函数声明.它在工具文件插入一个有框架工具的完整正确的函数定义.

8 按如下修改 OnMapMouseDown 以提供移动缩放能力并适当修改 mousepointer

9 重新编译 SuperMapDlg.cpp 和 BuidSupermap.exe

在运行应用程序时,如果你按左键拖动鼠标,你可以圈一个矩形来指定 Map 的一个区域来放大.如果你按右键,你可以左右移动 map.注意鼠标箭头的变化加入附加的控制我们已经加进了放大和移动 Map 的功能,然而我们还无法回到 map 的全范围.

加入一个附加控制

1 单击 ResouceView 按钮.

2 在对话文件夹中,双击 IDD--SUPERMAP--DIALOG.SuperMap 对话框出现.

3 选择 Button 控制(在 Controls 工具栏中)然后单击 MapControl 的右边把它安在对话框中.

4 右击这个按钮然后选择 Properties.在出现的 PushButtonProperties 对话框中,把标题改为 FULLEXtent

5 切换到 FileView 中并打开 SuperMap 文件夹.

6 双击 SuperMapDlg.cpp

7 在 WizardBar 中,在 ObjectIds 下拉列中选择 IDC--BVTTDN

8 在 Messages 下拉列中,选择 BN--CLICKEDAddMemberFunction 对话框出现.

9 单击 OK

你将位于成员函数中,在那你可按如下修改 OnButton1 来提供返回

map 全范围的功能.

一旦你完成了编辑,重新编译 SuperMapDlg.cpp 并重新建立可执行文件

来尝试这些改变.

总结:

在本章中你在 DeveloperStudio 中创建了一个基于对话的应用程序,你在其中插入了 Map 控制,你改变了控制的外部特征,改变了它的一些值,对它作了一些要求,如果你已读了本书的其它章节,你就已经知道函数性的广阔范围,都超出了你在本章中用的基本步骤,我们希望在你在 VisualC++环境中利用 MapObjects,本章能给你一些所需的初步指导.

第八章

第八章　配置应用程序

当你编完了应用程序,你也许就想把它分配给他人,把它配置在其它机器上.

大多数人选择创建一个特殊的 setup 程序来拷贝所有必要的文件和寄存所有适当的软件目标,你创建此程序所用的语言或应用程序全由你自己决定

通常 setup 展示了一个有选择和选项的友好菜单.你的应用程序所用成份类型不同.所需安装的程序文件和其它成份也不同,本章集中讨论 MapObjects 的安装要求,对于应用程序也需的其它成份和运行期库的安装要求不作累述.找到你的编程环境的参考文件,查看关于如何正确分配你的应用程序的细节.

当你分配应用程序时,你必须确保所有的可用的 OLE 成份被正确的寄存在目标系统的寄存数据库里,所有的支持文件都可以得到.这个任务会不会被自动执行取决于你的编程环境.

关于配置应用程序我们要讨论的几点内容是:

.OLE 的寄存.

.MapObjects 文件的要求.

.ODBC 文件的要求.

.使用 VisualBasic 的 setupWizard.

.关于应用程序的数据分配.

OLE 寄存的要求

MapObjectsOLECoponents 必须被正确的寄存在目标系统的寄存数据库里,你可以运行 REGSVR32.EXE 并把 DLL 文件成份的名称打在命令行上.REGSVR32.EXE 是 Microsoft 免费提供的程序.在 InstallData\Microsoft 下我们已把输送到了 MapObjectsInstall 盘上.你的 setup 程序也可用每个成份的函数 D11RegisterServer 来行使和 REGSVR.EXE 同样的功能.细节请看微软的 OLESDK 参考文件.下方标有*的文件需求对它运行看如何寄存 OLE 服务器和.OCX 文件.

MapObjects 文件的要求

你的 MapObjects 应用程序的功能不同,配置的文件组也就不同,如下所示.

注意:你绝不能分配 MapObjects.LIC 文件.分配这个文件将违反你和 ESRI 的许可证协议(LisenseAgreement).

核心成份

所有使用 MapObjects 的应用程序至少得重配重以下核心成份.DLL 文件.

SDE 软件成份

如果你的应用程序使用了 SDE,你必须也安装以下 SDE 成份:

SDE21.dll

SDE.dll*

图像成份

如果你的应用程序使用了图像层例如 TIFF 和 BMP,你也必须安装以下的图像成份:

AIImage.dll.*

DBC 文件要求

MapObjects 用微软开发数据库标准(ODBC)来连接和使用数据库.ODBC 和 MapObjects 驱动程序

必须正确的安装在目标机器上以让你的应用程序无误的运行.在此推荐下列程序来把 ODBC 和 MapObjects 的驱动安装并入你自己程序的驱动安装.

1 把 ODBCshape 文件夹所有内容拷贝到到一张 1.44MB 软盘上.

2 用你媒体的其余部分包含 ODBCShape 软盘.

3 确定在你要安装的名件列表中在 InstaallData\Microsoft\ODBC 下包括了 ODBC32.DLL 文件.

4 在你的安装指示中,指导用户在装完应用程序之后安装 ODBCShape 软盘

或者,你可用 ODBC.SDK 自己安装 ODBC.保证所有随 SHPDRIVER.DLL 的可重分配文件正确安装在目标机器系统的目录下.所有未压缩的必要的文件可以在 MapObjects 安装盘的 InstallData\Microsoft\ODBC 目录下找到.

TRUETYPE 字体

如果你的应用程序要利用提供 MapObjects 的 TrueTYPE 字体,例如 ESRI-Environment.ttf.这些字体必须被正确安装在目标机器上.此外,字体必须寄存在目标系统上.找到你的 setup 程序参考文件查看如何寄存字体的细节.你也可以查看 Win32SDK 参考文件中有关用 OS 功能执行这个操作的内容.

使用 MicrosoftVisualBasic 的 SetupWizard

如果你使用 MicrosoftBasic4.0 作为开发环境,你可以用 VisualBasic 的 SetupWizard 来产生一个 setup 程序.关于如何正确使用 VisualBasic 的 SetupWizard 的细节请看<<VisualBasicProgrammer'sGuide>>的第 30 章"分配你的应用程序".在以下用的 SetupWizard 的每一个步骤中,执行如下所示的特别操作,这些操作专用于创建使用 MapObjects 这类应用程序的 setup 程序.

1-3 步:

照 SetupWizard 指示做.

4 步(只当你的应用程序使用图像数据或 SDE).

1 单击 AddOLEServers.

2 定位并增加下列.DLL 文件:

AIImage.dll

Sde.dll

5 步

核对以下文件在相关文件列中被选上.

如果文件不存在,则 MapObjects 安装先于 VisualBasic 的安装.如果这样退出 VisualBasicSetupWizard 并重新安装 MapObjects 你将不得不重新运行 SetupWizard.

6 步:

选择"Installinapplicationdirectory"作为你的配置摸式.

7 步:

如果需要的活照 SetupWizard 的指示做.

SETUP.LST 示例

VisualBasic 的 SetupWizard 为名为 SuperMap.exe 应用程序创建的 Setup.Lst,这里是它的[FILE]部分的引述.注意你的应用程序可能会使用不同的文件.你的 DLL 和.OCX 文件的版本号也会同如下所示不同.

用你的应用程序分配数据

在很多情况下你将用你的应用程序提供数据.有关配置应用程序这方面最重要的内容包括如何满足用户何处安装地理图形数据的要求.用户们可能会选择定位的安装应用程序而不是缺省.所以你必须为你的安装程序提供一种方式来输送数据.使应用程序代码更容易以无缝的方式来使用它.在 MapObjects 的文件夹中所有的应用程序使用这样的模式.每个应用程序包含一个返回含所需数据文件夹的数据路径功能的模块.你也许要为你的应用程序建立一个相似的模式.下面是一个例子.

本密码认为地理数据存于与含有本应用程序的文件夹同级的 Data 文件夹中。该文件夹的 Usa(用户)文件夹包含应用程序要求的 GeoDataset