

# MapGIS 10.2 Objects 开发

## 入门向导

武汉中地数码科技有限公司

中国·武汉

2017 年 3 月

# 入门必读

尊敬的 MapGIS 二次开发用户：

您好，非常感谢您选择使用 MapGIS 平台软件进行开发。为帮助您更好地使用本手册，使您能够轻松、高效地进行 MapGIS 二次开发，特作此使用指南。希望本手册能够助您一臂之力！

## （一）模块说明

**产品介绍：**该部分通过平台介绍、平台开发框架和二次开发体系以及二次开发库等内容的介绍，帮助用户整体了解 MapGIS 10.2 Objects 二次开发平台产品与二次开发框架；

**环境准备：**该部分主要介绍基于 MapGIS 10.2 平台进行桌面 GIS 开发的环境要求，以及如何安装 MapGIS 10.2 平台与各种必备环境，指导用户配置开发环境；

**快速入门：**该部分作为 MapGIS 10.2 Objects 二次平台基于 C#开发的入门向导，指导用户动手开发第一个桌面 GIS 程序，帮助用户快速掌握二次开发的基本方法；

**常见问题：**该部分为 MapGIS 10.2 Objects 二次开发中的常见问题与解答，方便用户在遇到问题时能够第一时间解决。

## （二）使用向导

从未接触 C#桌面端开发，或者 GIS 开发的用户，建议根据实际情况先从开发基础参考了解相应的 C#桌面端的开发基础、GIS 相关基础知识，再通过产品介绍、快速入门进行学习。

入门之后，请通过《MapGIS 10.2 Objects 二次开发手册》进一步学习与演练。

## 目 录

<b>第 1 章 产品介绍</b> .....	1
1.1 平台开发框架 .....	1
1.1.1 二次开发特点 .....	1
1.1.2 二次开发架构 .....	1
1.2 二次开发库 .....	2
1.2.1 MapGIS 10.2 控件库 .....	2
1.2.2 MapGIS 10.2 方法库 .....	2
<b>第 2 章 环境准备</b> .....	3
2.1 开发环境要求 .....	3
2.1.1 二次开发环境要求 .....	3
2.1.2 开发者授权安装 .....	3
2.2 MapGIS 10.2 平台安装 .....	3
<b>第 3 章 快速入门</b> .....	8
3.1 C# 快速入门 .....	8
3.2 Objects 快速入门 .....	11
3.2.1 新建 Windows 窗体程序 .....	11
3.2.2 注册 MapGIS 地图显示控件 .....	11
3.2.3 添加必要的 MapGIS 程序集引用 .....	14
3.2.4 通过拖拽方式添加地图显示控件 .....	15
3.2.5 通过代码添加地图显示控件 .....	16
3.2.6 命名空间引用及变量定义 .....	17
3.2.7 实现打开地图文档功能 .....	18
3.2.8 运行程序 .....	19
<b>第 4 章 常见问题</b> .....	20
4.1 数据管理类 .....	20
4.2 数据表达类 .....	20

# 第1章 产品介绍

## 1.1 平台开发框架

### 1.1.1 二次开发特点

- MapGIS 10.2 提供了一套完整的且功能齐全的组件、控件及插件，用户只需要灵活地应用它们便可以非常方便简洁地开发各自行业领域内的 GIS 平台及应用系统；
- 易于集成，可以很容易与管理信息系统（MIS）、办公自动化（OA）以及专业应用模型有机结合，集成高效、无缝的 GIS 应用；
- 根据开发应用系统的规模大小，可以有目的地选择 GIS 组件和控件，有效地控制系统成本及风险；
- 多层次访问 MapGIS 组件的数据层、功能层、界面层，分层结构的划分使得组件之间关系更为清晰，逻辑更为明晰。

### 1.1.2 二次开发架构

面向桌面端应用开发，MapGIS 10.2 提供两套开发思路：其一是基于 MapGIS 基础的二次开发库，在 .NET Framework 框架上，构建您的应用系统，即 Objects 开发；其二同样基于 MapGIS 基础二次开发库，在 MapGIS 插件框架上，采用“框架+插件”模式构建您的应用系统，即插件式开发。MapGIS 10.2 C/S 二次开发架构如图 1.1 所示，本手册主要介绍 Objects 开发入门。

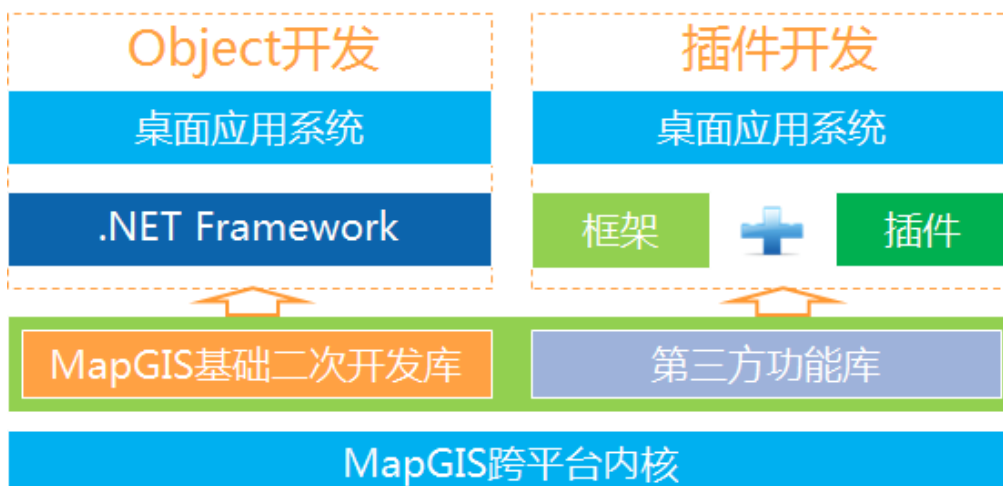


图 1.1 二次开发层次架构图

## 1.2 二次开发库

MapGIS 10.2 提供的二次开发库都是基于 .NET Framework 的程序集，必须有 .NET 框架的支撑才能运行。

### 1.2.1 MapGIS 10.2 控件库

MapGIS 10.2 提供了一套功能强大的带有 .NET Framework 控件的程序集，包括地图可视化控件、三维场景控件、MapGIS 基础工具控件等。下面是核心控件程序集列表：

图表 1-1 MapGIS 控件说明列表

程序集名称	说明
MapGIS.GISControl.dll	地图可视化控件，主要用于地图的显示，包括地图的放大、缩小、复位、平移等。
MapGIS.SceneControl.dll	三维场景控件，提供了三维场景显示、三维模型展示等三维数据表现的相关控件和方法。
MapGIS.UI.Controls.dll	MapGIS 基础工具控件，提供基础的工具窗口，例如空间参照系选择、颜色表选择等工具。

### 1.2.2 MapGIS 10.2 方法库

MapGIS 10.2 提供了一套功能全面的 .NET Framework 程序集，包括地图操作、查询、编辑空间分析以及数据转换等 GIS 功能，下面是核心组件库列表：

图表 1-2 MapGIS 组件说明列表

程序集名称	说明
MapGIS.GeoMap.dll	地图管理及可视化组件，主要提供了有关地图和图层相关的操作，以及坐标转换和自定义绘图方面的功能。
MapGIS.GeoObjects.dll	基本对象定义及管理组件，提供了大量基本对象的定义，比如要素、属性结构、属性等。
MapGIS.GeoDataBase.dll	矢量空间数据管理组件，提供了基本的类和数据集以及相关对象的定义及操作。
MapGIS.SpatialAnalysis.dll	主要用于进行空间运算和 buffer 分析以及拓扑查错等空间分析。
MapGIS.RasAnalysis.dll	主要用于进行栅格相关的分析，例如栅格裁剪、离散数据网格化、重分类等遥感专业分析接口。

## 第2章 环境准备

### 2.1 开发环境要求

#### 2.1.1 二次开发环境要求

MapGIS 10.2 的桌面 GIS 系统开发，对系统硬件环境没有特别要求，操作系统支持 Microsoft Windows 系列，包括 Win2000、Win Server2003（2008）、Win XP、Win Vista、Win7 等，以及 redHat、HP Unix、Linux、IBM AIX 等操作系统，均支持 32 位与 64 位机器。

一般需要依次安装配置下列软件环境：

- 集成开发环境：Microsoft Visual Studio(2005 或以上版本)，其中插件开发需使用 .NET Framework 4.0 及以上版本；
- MapGIS 平台许可：MapGIS 开发者授权；
- MapGIS 开发平台：MapGIS 10.2 桌面开发包；
- MapGIS 插件扩展开发向导包。

#### 2.1.2 开发者授权安装

必须在 MapGIS 开发者授权成功的状态下，才能正常安装、使用 MapGIS 产品，进行 MapGIS 二次开发。您可以通过访问司马云官方网站（<http://www.smaryun.com/>）获得开发者授权安装包。

### 2.2 MapGIS 10.2 平台安装

您可以通过 MapGIS 云开发世界下载软件安装包，离线安装配置所需开发环境，即分别下载 Microsoft Visual Studio 安装包、MapGIS 开发授权安装包、MapGIS 桌面开发包安装包以及 MapGIS 插件扩展开发向导安装包，同时还可以下载各个安装包相应的安装说明文档。

打开“云开发世界”→“资源中心”→“产品开发包”，桌面端开发环境的开发包下载界面如图 2 所示：



图 2.1 离线安装下载界面图

依次下载并安装上述安装包，根据说明文档进行安装。安装过程如下：

- (1) 安装 Micorsoft Visual Studio 2010，具体安装过程略；
- (2) 安装 MapGIS 开发授权，直接运行下载的开发授权文件（reg）即可；
- (3) 安装 MapGIS 桌面开发包，运行下载的安装包程序（exe），具体安装过程如下：



图 2.2 欢迎界面



图 2.3 类型选择



图 2.4 检测安装必备组件





图 2.5 选择安装平台的组件



图 2.6 安装过程



图 2.7 完成安装

## 第3章 快速入门

### 3.1 C#快速入门

在此章节,通过开发一个简单的 WindowForm 程序示例,让您尽快了解并熟悉 C#开发。具体实现该示例步骤为:

- (1) 在 Micorsoft Visual Studio 2010 (简称 VS2010) 中新建 Windows 窗体程序,界面如图 3.1 所示:

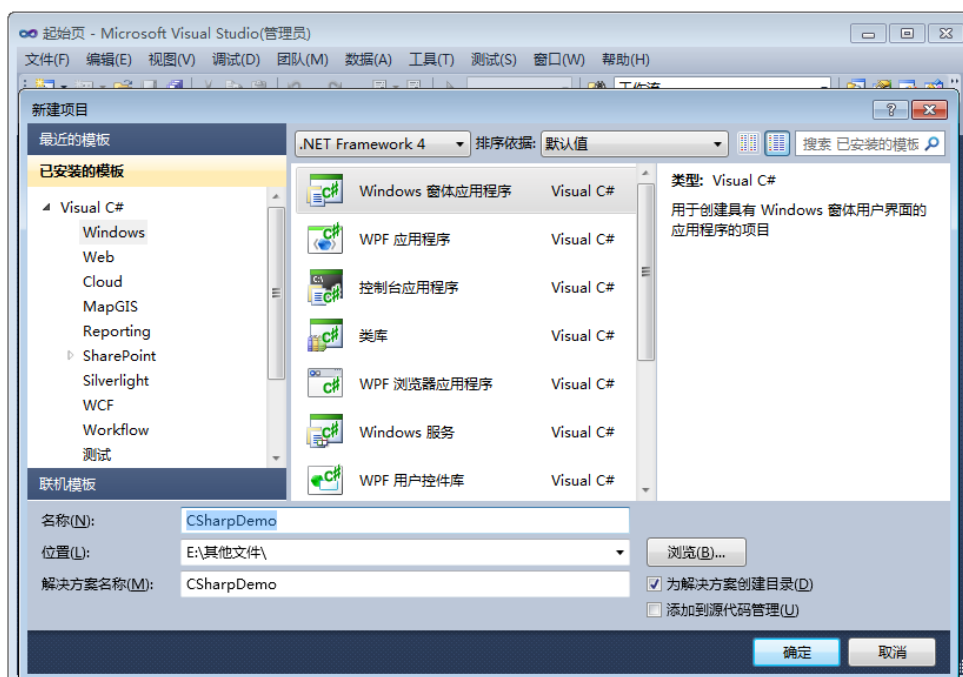


图 3.1 VS2010 新建项目界面

成功创建 CSharpDemo 程序之后如图 3.2 所示:

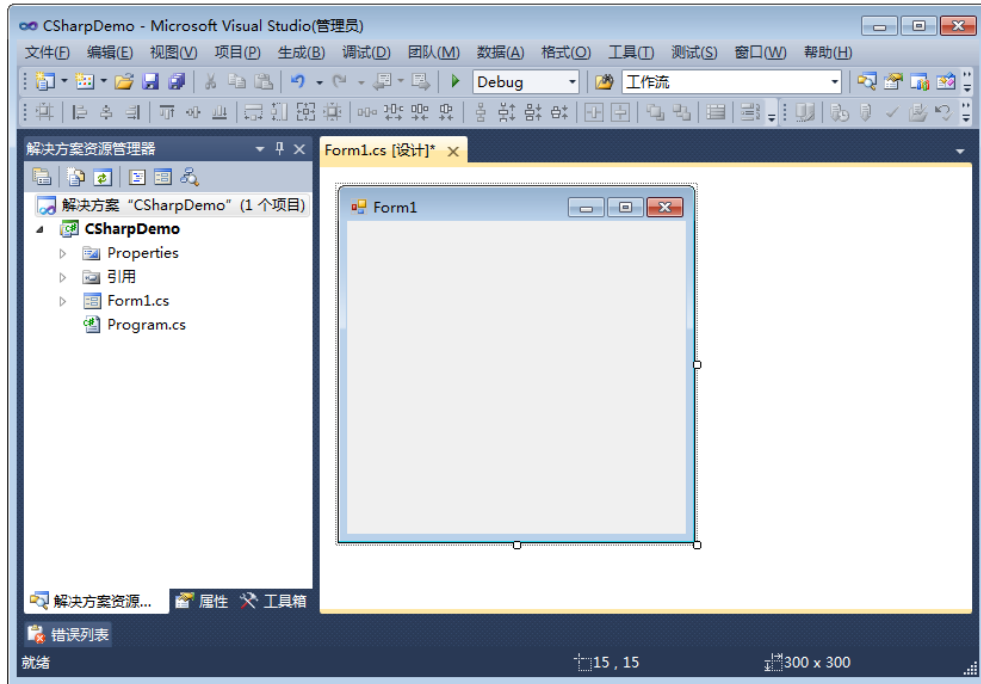


图 3.2 CSharpDemo 示例项目界面图

- (2) 调用 Button 按钮控件，从工具箱中调用按钮，在属性栏中修改该 Button 的 Text 属性值为“HelloWorld”，如图 3.3 所示：

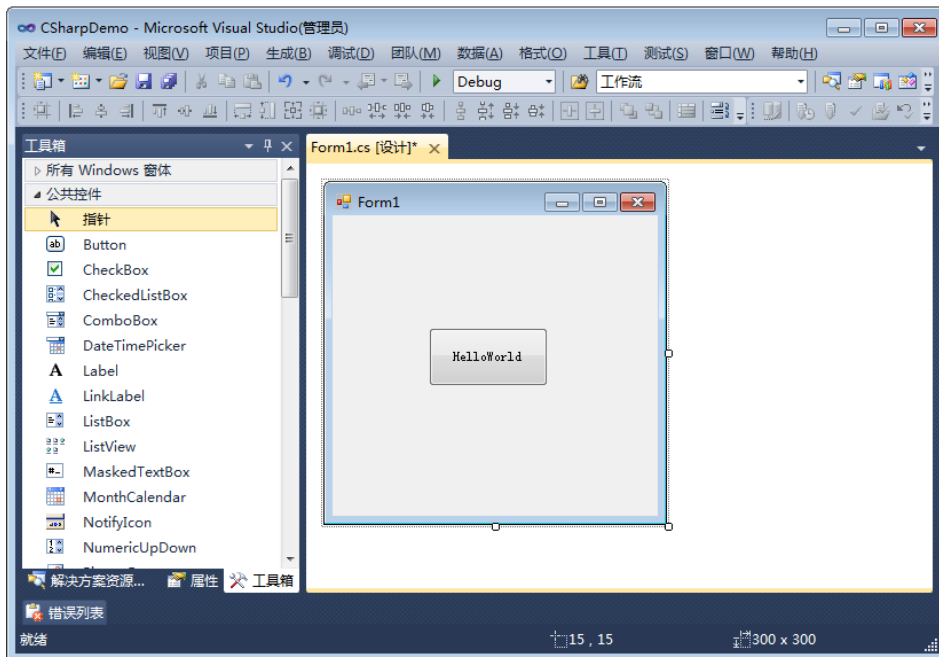


图 3.3 添加 Button 按钮界面图

- (3) 注册该按钮单击事件，编写消息输出代码，如图 3.4 所示：

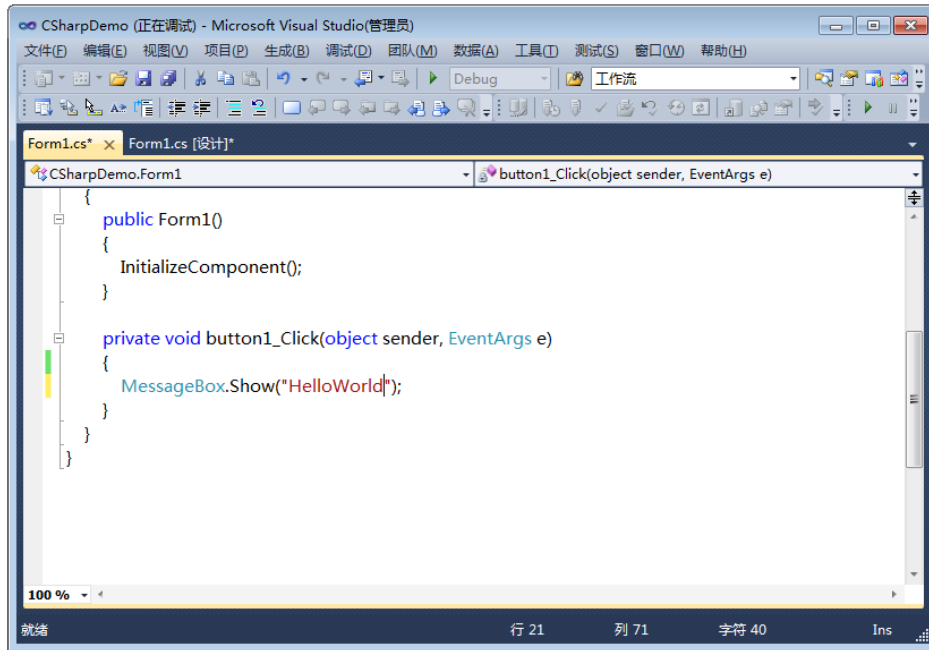


图 3.4 添加 Button 的 Click 事件代码图

(4) 运行上述创建的窗体程序，其效果如图 3.5、图 3.6 所示：

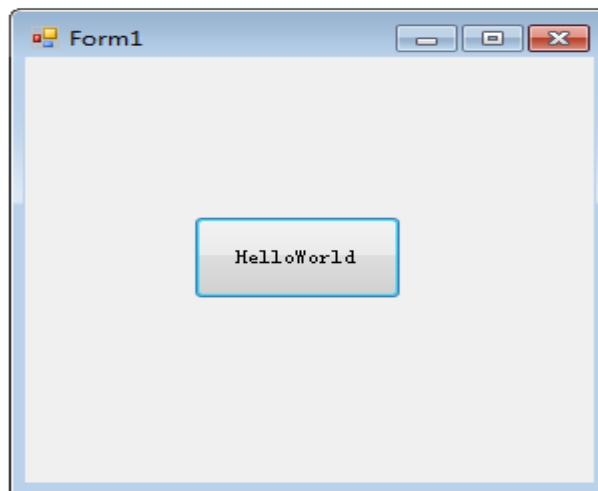


图 3.5 程序运行界面图



图 3.6 点击按钮响应界面图

## 3.2 Objects 快速入门

本小节将演示如何在 Microsoft Visual Studio 2010 中用一个 Winform 窗体中显示一幅 MapGIS 地图，帮助您快速进入和掌握 MapGIS 10.2 Objects 二次开发。具体实现步骤为：

### 3.2.1 新建 Windows 窗体程序

在 VS2010 中选择“文件→新建→项目”，然后在弹出框的左侧模板栏选择“Visual C#→Windows”，中间选择“Windows 窗体应用程序”，并设置“名称”、“位置”、“解决方案名称”等属性，新建项目界面如图 3.7 所示：

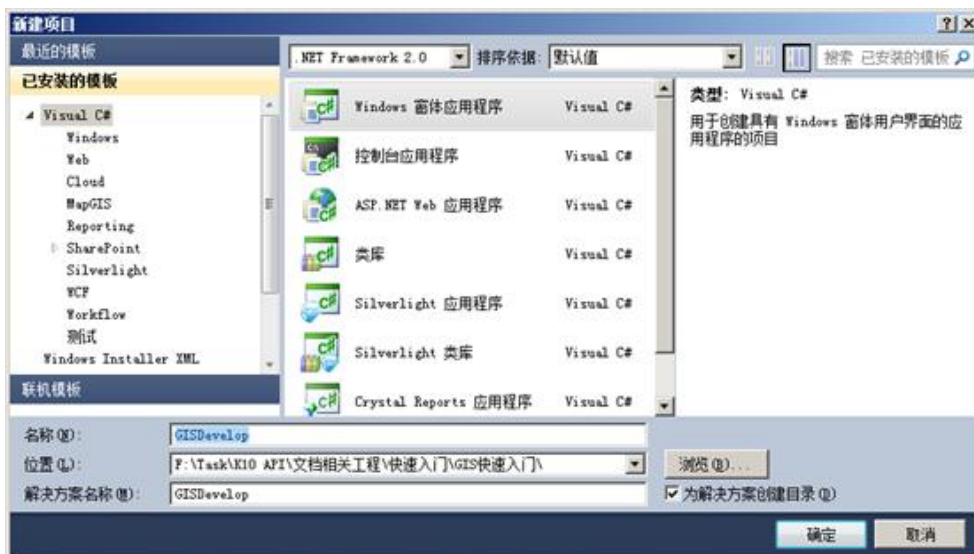


图 3.7 VS2010 新建项目界面图

### 3.2.2 注册 MapGIS 地图显示控件

地图显示控件由程序集“MapGIS.GISControl.dll”提供，该程序集位于 MapGIS 10.2 平台 Object SDK 目录（...\\MapGIS 10\\Object）中。因此，需要将此程序集里的地图显示控件注册添加到 Microsoft Visual Studio 2010 的工具箱中，以方便拖拽式调用该控件到 Form 设计界面。

注：若拖拽过程致使 VS2010 程序出现卡掉情况的话，需要将平台安装路径即...\\mapgis 10\\program 路径添加到环境变量中。

- (1) 打开“工具箱”面板，鼠标在“常规”选项卡上右击，选择“选择项”菜单，将地图显示控件添加到该选项卡下，如图 3.8、图 3.9 所示：

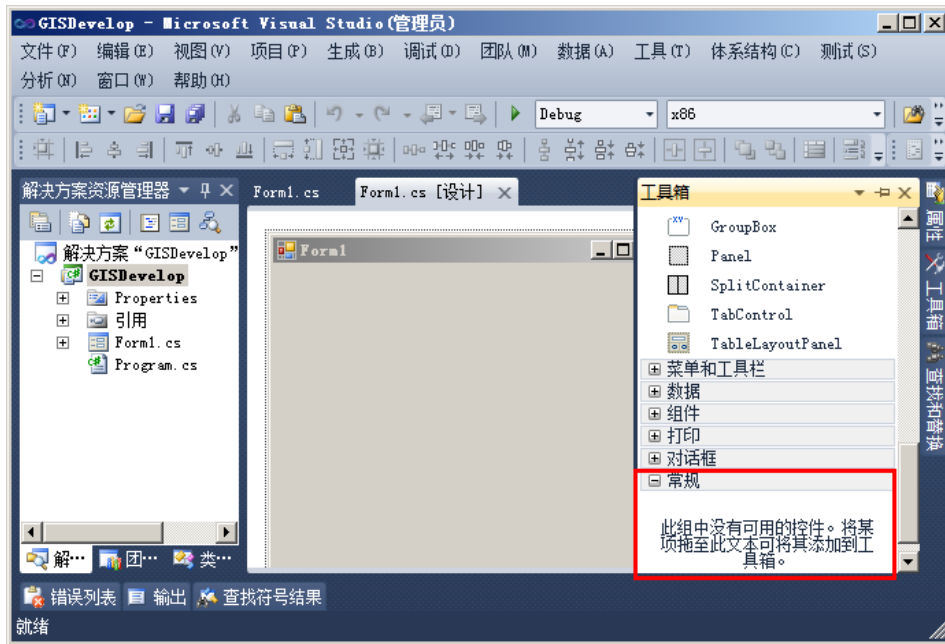


图 3.8 工具箱界面图

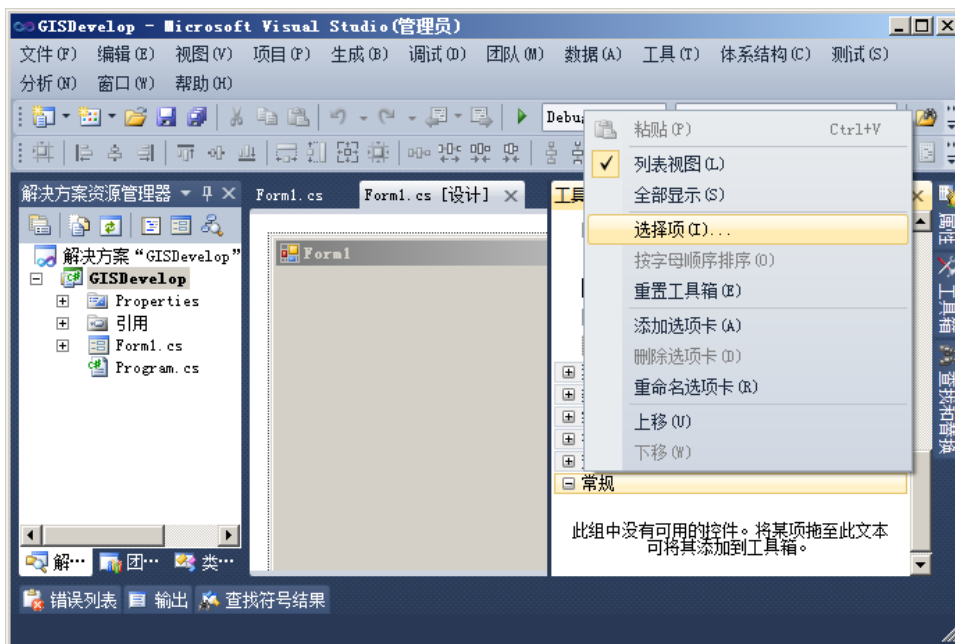


图 3.9 工具箱选择项界面图

注意：

(1)若您发现 Microsoft Visual Studio2010 中已有窗口中未发现上图右侧的工具箱面板，请到“视图——工具箱”调出该面板；

(2)若发现工具箱面板是空白状态，您需要切换上图的 Form1 窗体到设计视图模式，方法：右击上图左侧“Form1.cs”，选择“视图设计器”即可。

(2) 点击图 3.9 中的“选择项”，弹出窗口如图 3.10 所示：

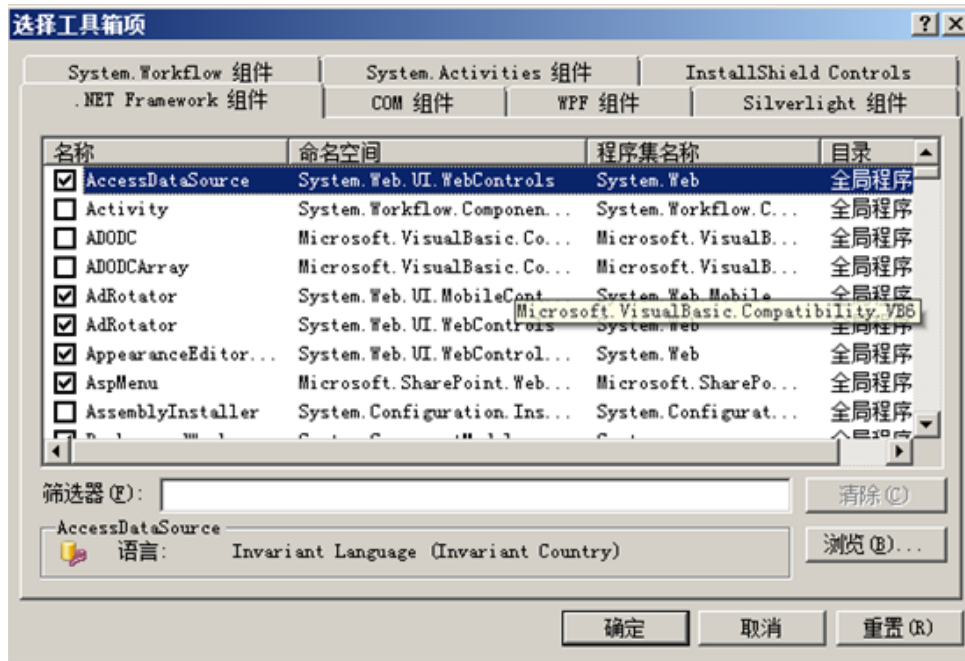


图 3.10 添加 Framework 组件界面图

- (3) 通过单击图 3.10 中的“浏览”按钮，找到 MapGIS 10.2 平台安装目录下的程序集“MapGIS.GISControl.dll”，然后单击确定按钮加载该程序集；此时，该程序集提供的相关 MapGIS 界面控件（包括地图显示控件）将列举在上图列表中，并默认处于勾选状态，如图 3.11 所示：

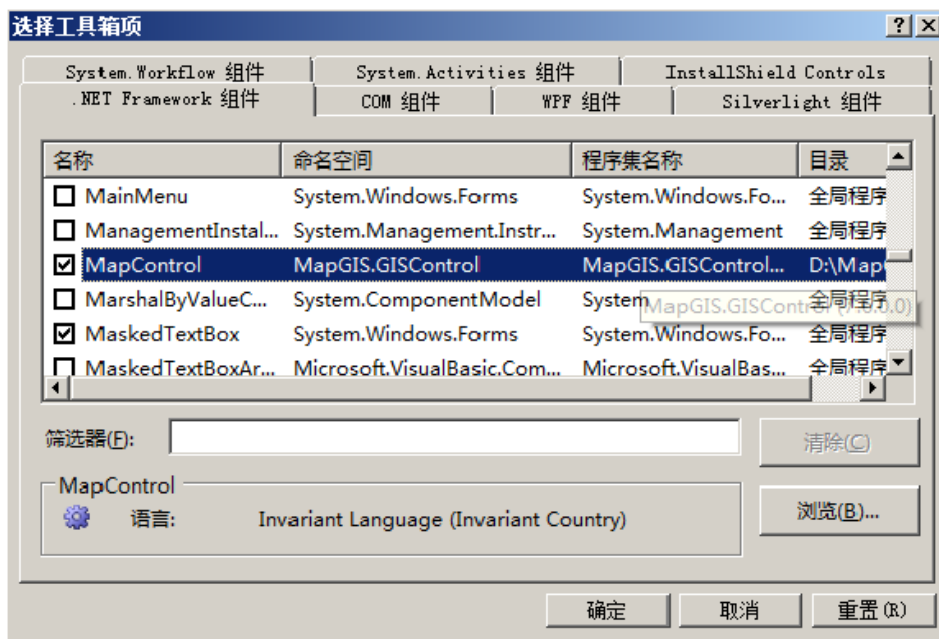


图 3.11 选择 MapControl 界面图

- (4) 图 3.11 默认勾选地图显示控件“MapControl”，单击“确定”按钮，添加此控件到工具箱里的“常规选项卡”类别下，如图 3.12 右下角所示：



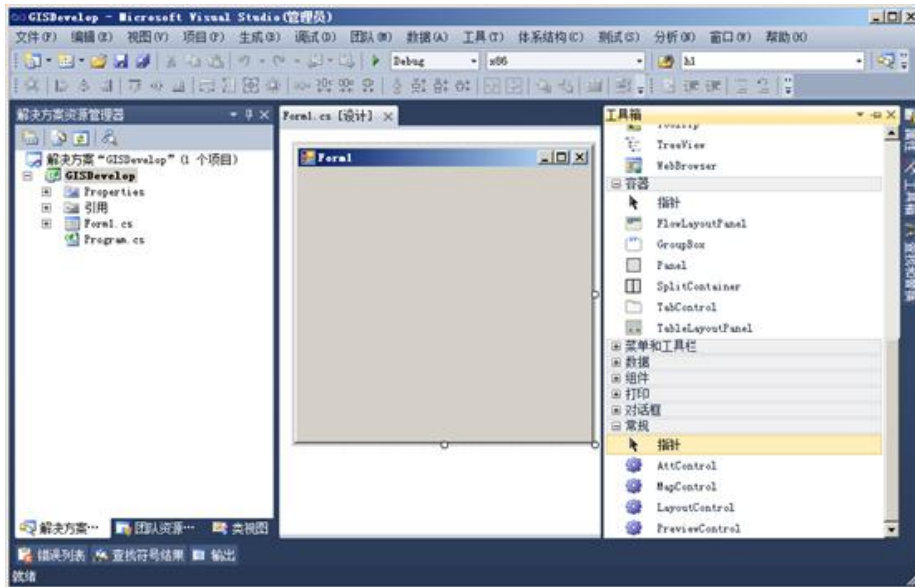


图 3.12 添加 MapControl 成功的界面图

到此为止，显示一幅地图所需的地图显示控件加载完成，接下来将实现具体的功能！

### 3.2.3 添加必要的 MapGIS 程序集引用

为项目添加必要的 MapGIS 程序集引用，在此需引用 MapGIS.GeoMap.dll、MapGIS.GISControl.dll。引用方法步骤为：右击解决方案“GISDevelop”项目下的“引用”选项卡，弹出如图 3.13 所示窗口：

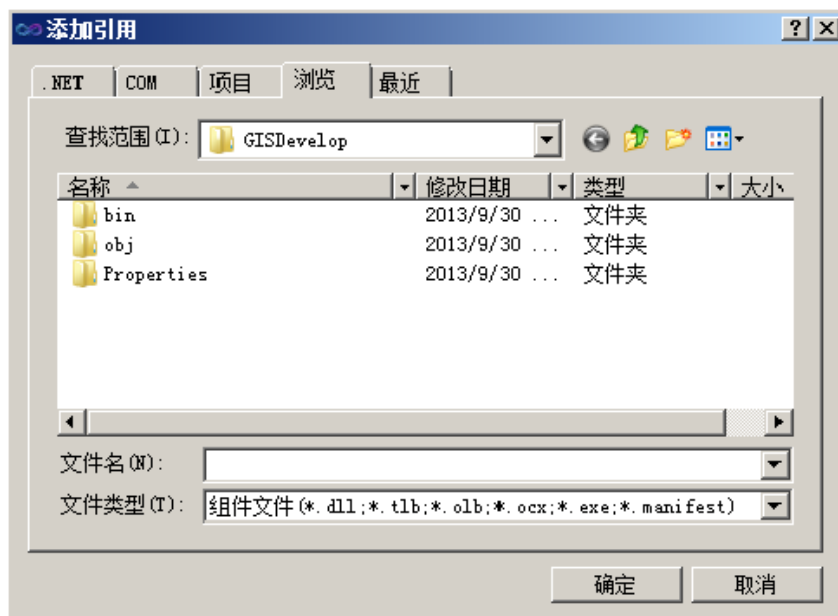


图 3.13 添加 MapGIS 组件界面图

切换到“浏览”选项卡，从 MapGIS 10.2 平台 Object SDK 目录（...\\MapGIS 10\\Object）找到上述两个程序集，单击“确定”按钮即可。实现程序集引用后如图 3.14 所示：

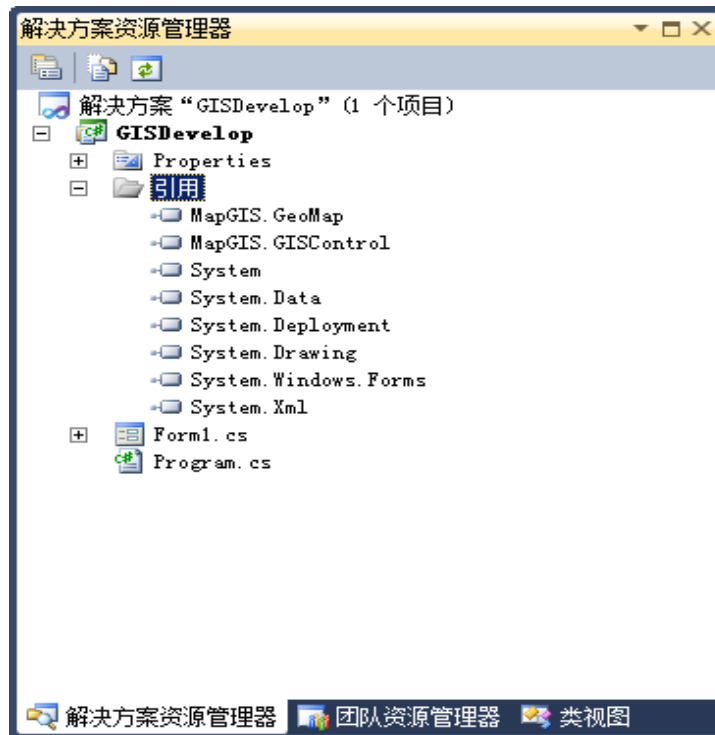


图 3.14 添加 MapGIS 程序集图

### 3.2.4 通过拖拽方式添加地图显示控件

通过鼠标拖拽的方式，从工具箱的常规选项卡下将第二步注册的地图显示控件（MapControl）拖拽到 Form1 设计视图，并调整好布局，如图 3.15 所示：

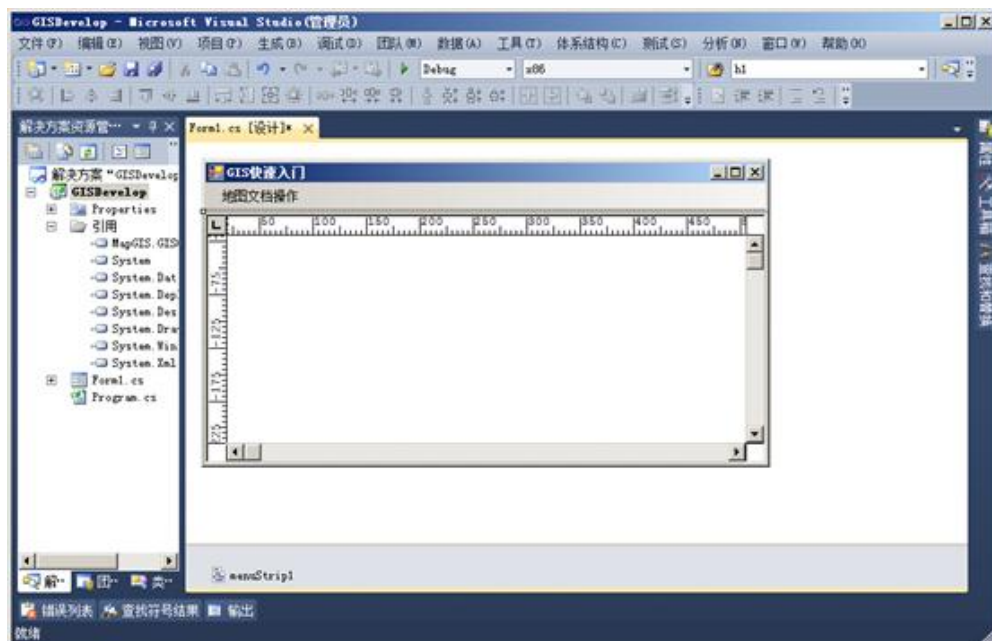


图 3.15 界面布局图

### 3.2.5 通过代码添加地图显示控件

地图显示控件由程序集“MapGIS.GISControl.dll”提供，在写代码之前，需添加该引用，如图 3.16 所示。

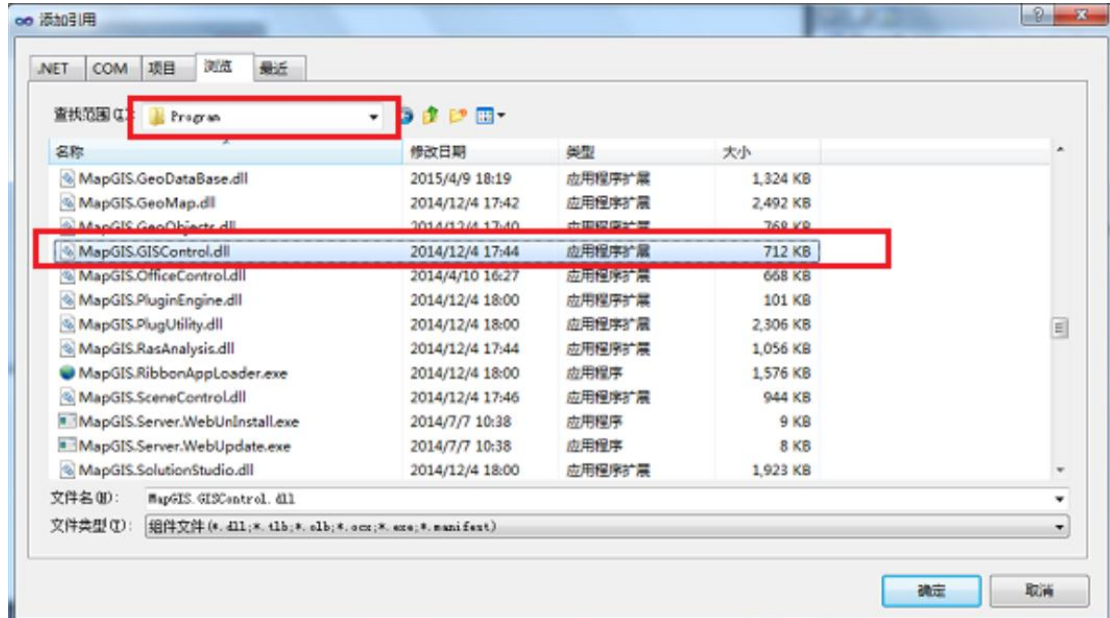
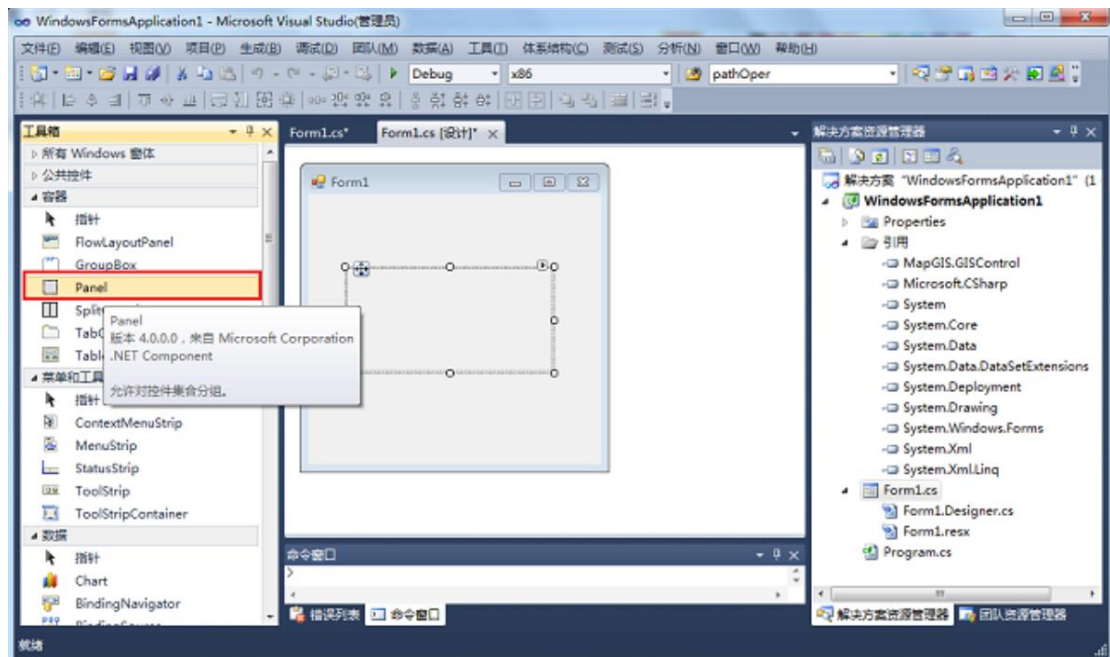


图 3.16 添加 GISControl 程序集

引用添加成功后，即可进入 Form 的代码页面，首先添加命名空间，然后创建地图显示控件（MapControl），并添加到 panel 里。这里，先在 Form 设计页面，拖入一个 panel，用来放地图显示控件。



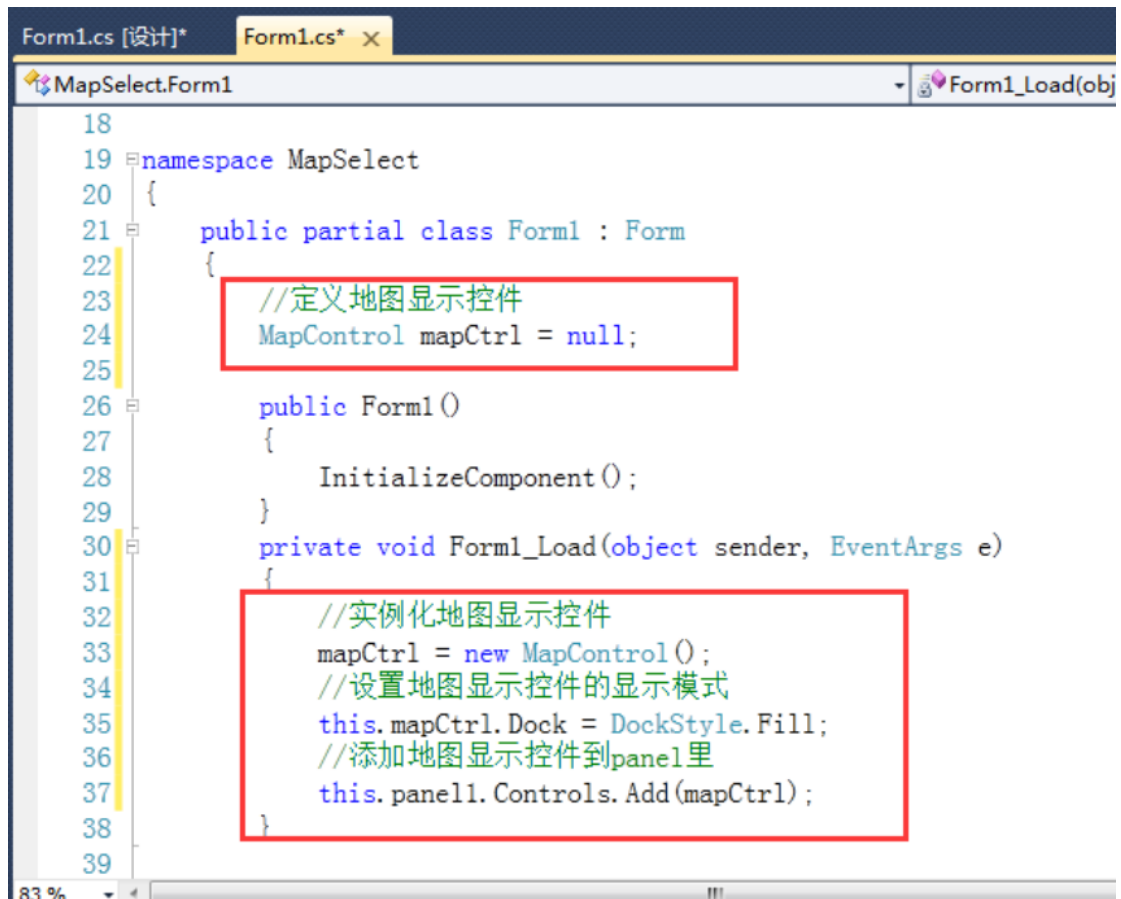


图 3.17 添加并绑定 panel

至此，已实现通过代码添加地图显示控件。

### 3.2.6 命名空间引用及变量定义

打开 Form1 的代码界面，分别添加程序集 MapGIS.GeoMap.dll、MapGIS.GISControl.dll 的命名空间引用，如下代码所示：

程序代码 3.1 命名空间引用关键代码

```

using MapGIS.GeoMap;
using MapGIS.GISControl;

```

接着，在类中添加变量定义的代码，如下代码所示：

程序代码 3.2 变量定义关键代码

```

//地图文档对象
Document doc = null;
//地图对象
Map map = null;

```

命名空间引用及变量定义代码添加完成后，如图 3.18 所示：

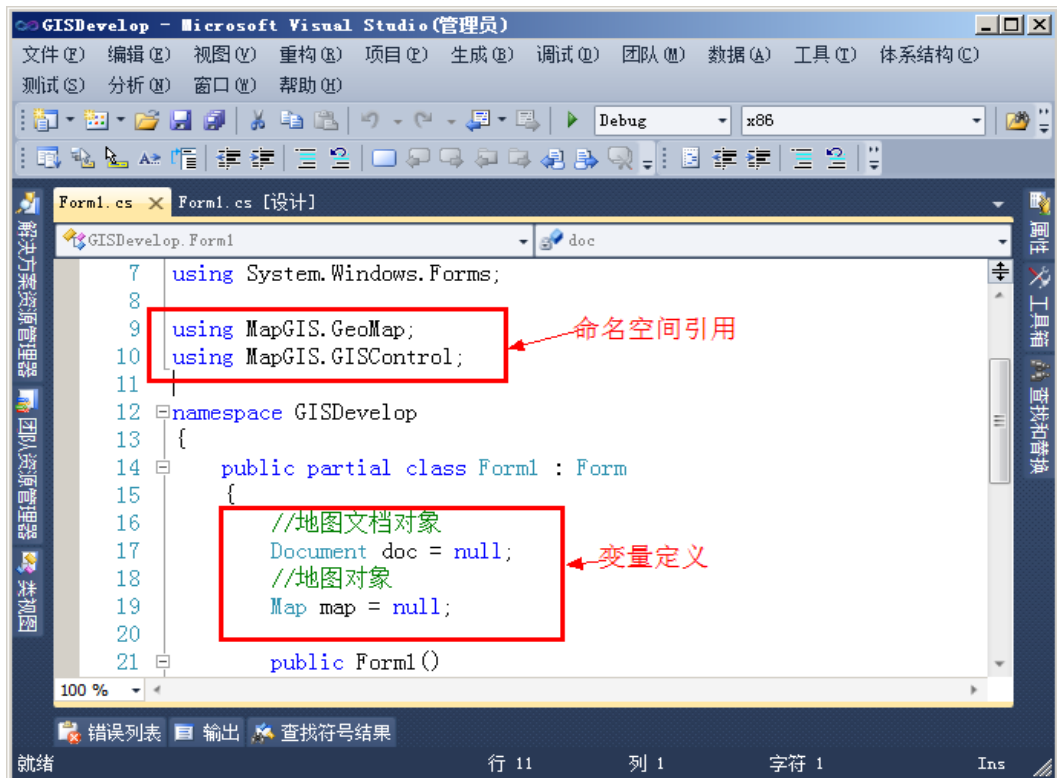


图 3.18 Form1 的代码界面图

### 3.2.7 实现打开地图文档功能

实现打开地图文档功能，具体步骤如下：

- (1) 在 Form1 视图设计窗口，添加功能菜单“地图文档操作”及其子菜单项“打开”，并为该菜单项“打开”注册单击事件处理函数；
- (2) 双击该“打开”菜单项即可自动实现单击事件处理函数的添加，程序自动跳转到相应代码界面；在已添加的单击事件处理函数里添加如下所示的打开地图文档的代码：

程序代码 3.3 消息处理关键代码

```

//打开地图文档
doc = new Document();
doc.Open(@"D:\MapGIS 10\Sample\世界地图.mapx");
//获取第一个地图
map = doc.GetMaps().GetMap(0);
//显示地图
this.mapControl1.ActiveMap = map;
//复位窗口
this.mapControl1.Restore();

```

注意：必须将地图文档中的图层对应的类所在的数据库加载到本地数据源下，否则打开地图文档在地图视图中显示不成功。

### 3.2.8 运行程序

打开地图文档功能实现后，点击“调试→启动调试”，程序运行效果如图 3.19 所示：

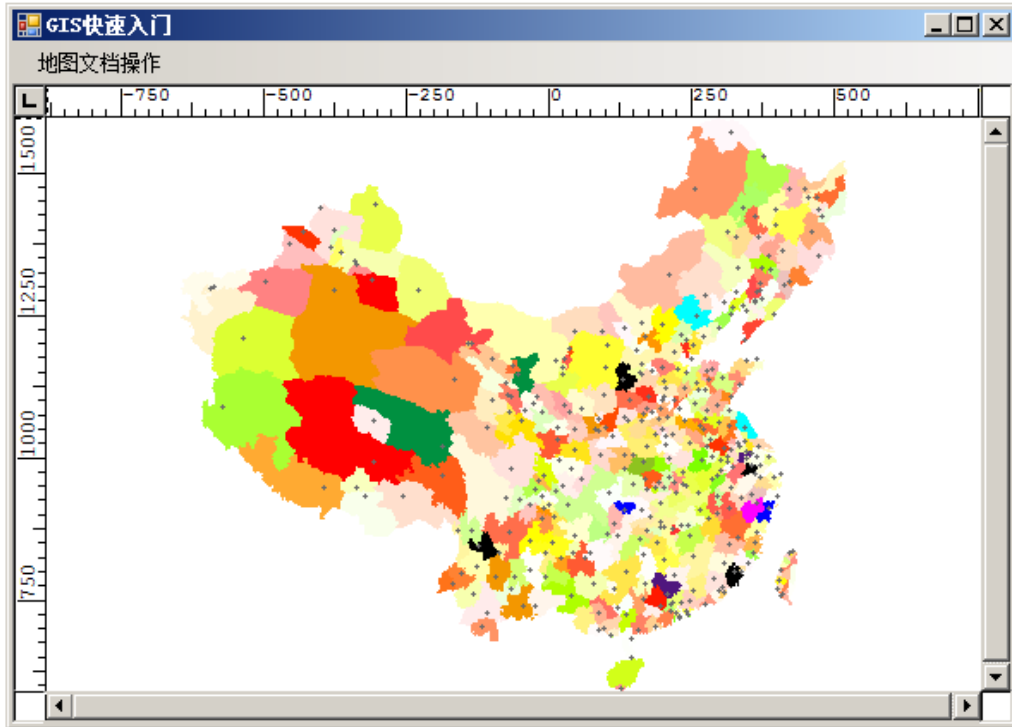


图 3.19 程序运行效果图

## 第4章 常见问题

### 4.1 数据管理类

#### 1. 操作矢量类时，报操作对象内存不可读，是什么原因？

答：原因一：在这之前 open 了图层，没有关闭导致的错误，在程序中对数据操作完成后要及时关闭数据；原因二：此数据在平台或其他程序正在进行编辑。

#### 2. 表格数据转换为简单要素类失败，是什么原因？

答：表格文件只能与对象类相互转换。

### 4.2 数据表达类

#### 1. 用 Display 自定义绘制图形，为确保刷新窗口亦能正常显示图形，应该怎么做？

答：应该将自定义绘制图形的方法添加到 MapControl 控件的 postrefresh 事件中即可，每次重绘地图（刷新、缩放、移动等引发）均会自动调用该方法，实现自定义图形重新自动绘制。

#### 2. Display 里的绘图方法基本都有两种方法可以实现，例如绘制多边形有方法 DispPolygon 和 Polygon，请问有什么区别？

答：一般带图形参数的方法是负责绘制逻辑坐标下的几何对象，而不带图形参数的则是绘制设备坐标下的几何对象，例如 DispPolygon(GeoPolygon poly, RegInfo regInfo)，需要传入图形参数 regInfo，则该方法所需的是逻辑坐标下的几何对象 poly，而方法 Polygon(GeoPolygon poly)，则是设备坐标下的几何对象 poly。

#### 3. MapControl 有很多窗口事件，例如 PostRefresh、BeginDrawing 等，用 Display 自定义绘图时会触发这样的事件吗？

答：自定义绘图时不会触发窗口重绘类型的事件。

#### 4. 用 Display 的 Polygon 方法始终画不出想要的矩形图形，可能是什么原因？

答：可能主要是两种原因导致，其一：矩形对象构建需要把第一个点作为第五个点添加到线对象中，以封闭矩形；其二可能是没有设置画刷导致。参考关键代码如下：

程序代码 4.1 绘制多边形关键代码

```
GeoPolygon polygon = new GeoPolygon();
GeoVarLine varLine = new GeoVarLine();
//构建折线对象（矩形）
varLine.Append(new Dot(100, 100));
varLine.Append(new Dot(200, 100));
varLine.Append(new Dot(200, 200));
```

```

varLine.Append(new Dot(100, 200));
varLine.Append(new Dot(100, 100));//以封闭矩形
GeoLines lines = new GeoLines();
lines.Append(varLine);
polygon.Append(lines);
//绘制该矩形
disp.Begin();
disp.SetBrush(6,0,0,0,0);//这个必须设置
disp.Polygon(polygon);//绘制多边形
disp.End();

```

### 5. 什么是异或模式？

答：异或模式是用 Display 类对象进行自定义图形绘制时可能用到的一种画笔模式，通过给定 SetPenMode 方法里参数值为 7 来设置当前画笔为异或模式，处于异或模式的画笔，将可实现：在同一位置，重复绘制同一几何对象（包括构成几何对象的所有坐标点及几何对象的角度完全一样）时，将实现擦除原几何对象的效果。

### 6. 如何显示部分动态注记？

答：这个问题涉及到两个方面：动态注记与部分显示。在 GeoMap 提供的图层对象（VectorLayer）里面都有相应的功能接口与之对应，动态注记由图层的属性 Label 提供；而实现部分显示则通过图层设置显示过滤条件，由函数 SetDispCondition 提供。示例代码如下：

程序代码 4.2 显示部分动态注记关键代码

```

//地图显示控件
MapControl mapControl1 = null;
//数据库服务器
Server srv = null;
//数据库
DataBase db = null;
//数据
SFeatureCls SFeature = null;
Map map = null;
VectorLayer layer = null;
//初始化控件
this.mapControl1 = new MapControl();
this.splitContainer1.Panel2.Controls.Add(this.mapControl1);
//打开数据源
srv = new Server();
srv.Connect("mapgislocal", "", "");
db = srv.OpenGDB("world");
//打开用于显示的简单要素类

```



```
SFeature = (SFeatureCls)db.GetXClass(XClsType.SFCls);
SFeature.Open("行政区.WP", 0);
layer = new VectorLayer(VectorLayerType.SFclsLayer);
if (SFeature != null)
    layer.AttachData(SFeature);
    //添加图层
    map = new Map();
    map.Append(layer);
    //显示图层
    this.mapControl1.ActiveMap = map;
    SimpleLabel splbl = null;
    //注记信息
    LabelInfo labelInfo = null;
    //注记图形信息
    AnnInfo annInfo = null;
    //区标注分布信息
    RegionPlaceInfo regPlaceInfo = null;
    splbl = new SimpleLabel();
    labelInfo = new LabelInfo();
    TextAnnInfo txtAnninfo = new TextAnnInfo();
    regPlaceInfo = new RegionPlaceInfo();
    #region 设置注记图形信息
    txtAnninfo.Color = 5;
    txtAnninfo.Height = 10;
    txtAnninfo.Width = 10;
    //间隔
    txtAnninfo.Space = 2;
    //压盖
    txtAnninfo.Ovprnt = true;
    #endregion
    #region 设置注记显示参数信息
    labelInfo.AnnInfo = txtAnninfo;
    labelInfo.Caption = "名称";
    labelInfo.IsVisible = true;
    labelInfo.MaxScale = 0;
    labelInfo.MinScale = 0;
    #endregion
```

```

#region 区标注分布信息
//每个要素一个标注
regPlaceInfo.DuplicateType = DuplicateType.OneLabelPreFeature;
//自适应
regPlaceInfo.LimitLabelSmallRegion = true;
//最大面积
regPlaceInfo.SmallRegionMaxArea = 500;
//可标注在区外
regPlaceInfo.TryLabelOutside = true;
//区方位-水平标注
regPlaceInfo.Type = RegPlaceType.HorizationPlace;
#endregion

#region 设置简单注记
//设置属性字段*
splbl.LabelExpression = "名称";
splbl.HighlightClr = 6;
splbl.LabelGeomType = LabelGeomType.RegionGeom;
//设置区标注分布
splbl.RegPlaceInfo = regPlaceInfo;
//高亮显示未能标注的注记
splbl.HighlightLabelUnPlaced = true;
splbl.LabelUnPlaced = true;
splbl.Priority = 1;
//显示动态注记
splbl.Visible = true;
splbl.SetInfo(labelInfo);
#endregion

//初始化条件显示条件
MapGIS.GeoDataBase.QueryDef disCondition = new QueryDef();
disCondition.Filter="ID>5";
disCondition.CursorType = SetCursorType.ForwardBackward;
//设置条件显示
layer.SetDispCondition(disCondition);
//设置动态注记
layer.Label = splbl;
this.mapControl1.Refresh();

```

## 7. 线型库中如何唯一确定一个线型？

答：通过组件方式只能操作系统默认的系统库，一个线型可以通过线型号+辅助线型号来唯一确定，其中：

- LinInfo.LibID: 辅助线型号
- LinInfo.LinStylID: 线型号

#### 8. 如何设置图层透明显示？

答:VectorLayer 图层对象提供了 Transparency 属性：0-100，值越大，图层显示效果越透明。

#### 9. 如何取得 MapGIS 颜色号的 RGB 值？

答：该功能由 MapGIS.GeoMap 提供，具体实现代码如下：

程序代码 4.3 获取指定颜色的 RGB 值关键代码

```
//系统库管理
SystemLibrary sysLibs = SystemLibrarys.GetSystemLibrarys();
if (sysLibs != null)
{
    //获取默认系统库
    SystemLibrary sysLib = sysLibs.GetDefaultSystemLibrary();
    if (sysLib != null)
    {
        //获取颜色库
        ColorLibrary mcLib = sysLib.GetColorLibrary();
        if (mcLib != null)
        {
            //颜色对象
            MapGIS.GeoMap.Color mClrBegin;
            //获取 45 号色的 R、G、B 分量
            byte rBegin, gBegin, bBegin;
            //根据颜色号找到颜色对象
            mClrBegin = mcLib.GetColorByNo(45);
            mClrBegin.GetRGB(out rBegin, out gBegin, out bBegin);
        }
    }
}
```

#### 10. 为什么添加弧段不成功？

答：在添加圆、弧以及 Bezier 之类的解析线时，添加不成功很多情况下都是由于构造完之后没有离散化，也就是没有调用 DisperseToDots，因为这些图形在存储的时候也是存储的点；再者添加不成功还有一个原因就是调用了 DisperseToDots()，但是入口参数步长给的不合适，可能太大了，这个步长是地图坐标，是根据你实际的地图数据而确定的，如果太小了，数据量会变大；如果太大了，可能会离散不成功；在添加区块的时候也存在着同样的问题，

可以参照如下代码：

程序代码 4.4 添加弧段关键代码

```
//地图控件
MapControl mapControl1 = new MapControl();
this.splitContainer1.Panel2.Controls.Add(mapControl1);
Display disp = mapControl1.Display;
GeoArc arc = new GeoArc();
//初始化弧段对象
arc.Set(new Dot(100, 100), 130, 45, 90);
//离散化
arc.DisperseToDots(2);
//开始绘制
disp.Begin();
disp.SetPen(1, 6);
disp.MoveTo(new Dot(100, 100));
//绘制弧段
disp.Arc(arc);
disp.End();
```

**11. 如何控制一个矢量类图层中某些要素在地图视图中可见或不可见？**

答：可以在 MapGIS.GeoMap 命名空间中 VectorLayer 类中的 SetDispCondition 方法，通过设置一定条件，控制当前图层的某些对象显示与隐藏。