

# 实验一 VisualBasic 基础

## 一、实验目的

了解 VisualBasic 基本界面、特点，掌握开发 VisualBasic 应用程序的基本步骤。

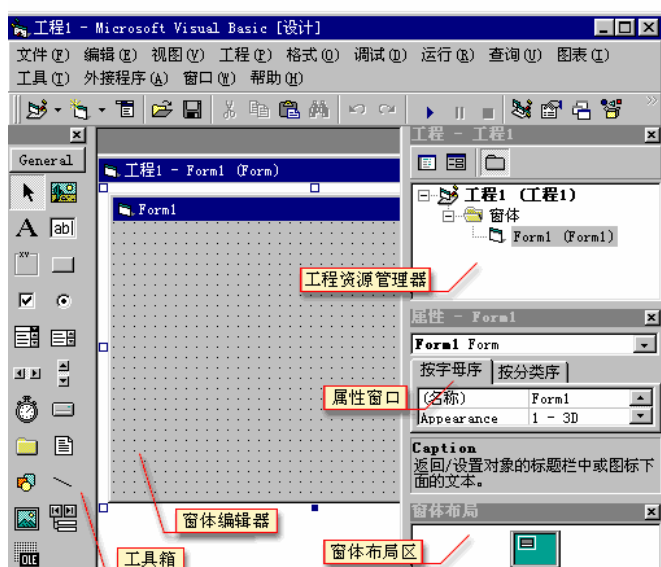
## 二、实验任务

- 1、掌握属性窗口、窗体设计器、代码窗口的主要功能；
- 2、通过例子掌握开发 VB 应用程序的基本步骤。

## 三、实验内容

### (一) VisualBasic 集成开发环境

VisualBasic 系统为用户开发应用程序提供了一个良好的集成开发环境，如下图所示。它集成了各种不同的功能，例如，用户界面设计、代码编辑、模块的编译、运行、调试等。该界面由多个窗口构成了 VB 的集成开发环境。开发 VisualBasic 应用程序时，需要将这些窗口配合使用。



### 1、主窗口

VB 系统的主窗口由标题栏、菜单栏和工具栏组成，为用户提供了开发 VB

应用程序的各种命令和工具。

### (1) 标题栏

标题栏与 Windows 其他应用程序的相似，显示当前的工程文件名和 VB 的当前工作模式状态，其中的[设计]表明集成开发环境此时处于设计模式。VB 的三种工作模式及其作用如下。

①设计模式。在该模式下，用户可以进行程序界面的设计和代码的编写工作。程序运行后出现错误时，对程序代码的编辑过程，也在设计模式中进行。

②运行模式。程序界面和代码的设计完成后，运行应用程序时处于该模式。VB 应用程序运行后，一直处于等待事件发生的状态中，退出应用程序，则回到设计模式。运行阶段不能进行界面和代码的编辑工作。

③中断模式。应用程序运行出现错误时，处于中断模式。该阶段可以编辑代码，重新运行程序，但是程序界面不能够被编辑。

### (2) 菜单栏

菜单栏中包含 VB 系统所有的可用命令，这是程序开发过程中用于设计、调试、运行和保存应用程序所需要的命令，共有 13 个菜单项。

[文件]: 主要提供对工程文件的创建、打开、保存和增删等操作命令。将当前应用程序生成可执行文件的操作命令也在其中。

[编辑]: 主要提供对应用程序进行编辑的各种操作命令。

[视图]: 主要提供设计程序界面、运行和调试程序时各种窗口的切换。

[工程]: 主要提供为工程添加窗体、模块、控制、部件等对象的命令。

[格式]: 主要提供对窗体控制的对齐、尺寸及间距等格式化的命令。

[调试]: 主要提供调试程序的各种命令。

[运行]: 主要提供程序启动、中断和停止等命令。

[查询]: 主要提供对数据库查询的相关命令。

[图表]: 主要提供对图表的新建、设置、添加、显示和修改等命令。

[工具]: 主要提供添加过程、过程属性、菜单编辑器、相关选项的设置及应用程序的发布等命令。

[外接程序]: 主要提供在 VB 中进行数据库管理和外接程序管理器的功能。

[窗口]: 主要提供窗口的排列和过程文件的切换命令。

[帮助]: 启动帮助系统, 打开帮助窗口, 为用户提供学习和查询 VB 的相关信息。

### (3) 工具栏

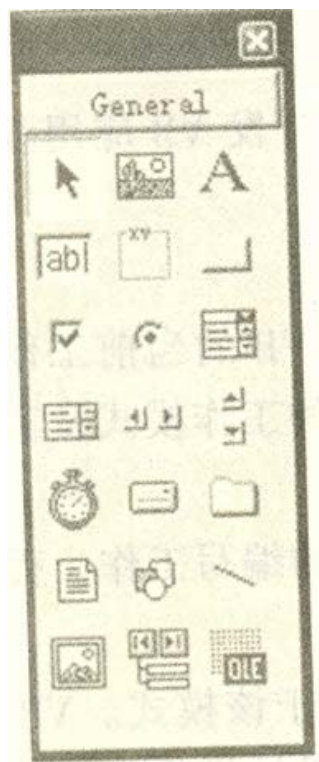
工具栏中集中了各种用图标表示的按钮, 每个按钮对应一个命令, 单击按钮, 即可执行对应的命令。默认情况下, VB 启动后显示标准工具栏, 除此, VB 还提供了编辑、窗口编辑器和调试等专用的工具栏。可以通过选择[视图]→[工具栏]命令将其他工具栏在这个集成环境中移入或移出。

### (4) 控件工具箱

控制部件简称控件, 是用户设计应用程序界面的工具。VB 的标准工具箱包含建立应用程序所需的各种控件, 如下图所示。另外, VB 还提供了很多 ActiveX 控件, 可以将它们添加到工具箱中。如果控件工具箱在集成环境中没有出现, 可以选择[视图]→[工具箱]命令使其再现。

标准工具箱中包含的控件, 按照下图的顺序依次为: 图片框 (PictureBox)、标签 (Lable)、文本框 (TextBox)、框架; (Frame)、命令按

钮(CommandButton)、复选框(CheckBox)、选项;按钮(OptionButton)、下拉式列表框(Combomox)、列表框(ListBox)、水平滚动条(HScrollBar)和垂直滚动条(VScrollBar)、定时器(Timer)、驱动器列表框(DriverListBox)、目录列表框(DirListBox)、文件列表框(FileLstBox)、轮廓(Shape)、直线(Line)、图像框(Image)、数据控件(Data)和 OLE 容器(OLE)等。



控件工具箱

工具箱作为用户开发应用程序工具的集合,用户对其使用方法掌握得越多,处理问题的能力越强,可以向工具箱添加自己需要的控件。但是,工具箱并非越大越好,系统允许用户自己定义一张"选项卡",向其中添加自己常用的控件,如下图所示。



自定义控件工具箱

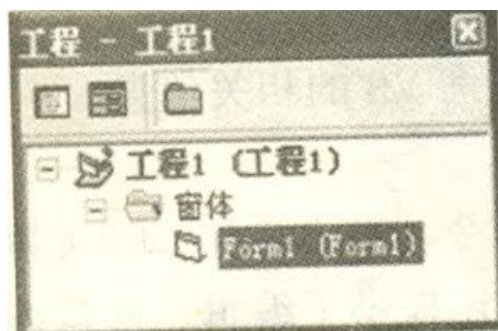
自定义选项卡及添加控件的操作如下:

- ①在工具箱的空白处,单击鼠标右键,在快捷菜单中选择[添加选项卡]命令。
- ②在打开的对话框中输入选项卡的名称,如 UDF。
- ③用鼠标将所需的控件拖动到选项卡 UDF 上。
- ④重复第③步可以添加多个控件;重复以上过程可以添加工具箱多个选项卡。

用户选项卡一经定义,可以随 VB 系统保存,供经常使用,不需要时可以删除。

## 2、工程窗口

工程是应用程序各种类型文件的集合,工程文件的扩展名为.Vbp。它包含的三类主要文件为:窗体文件(.Frm)、标准模块文件(.Bas)、类模块文件(.Cls)。工程文件就是与该工程有关的所有文件和对象的清单,这些文件和对象自动链接到工程。每个工程中的对象和文件也可以供其他工程使用。



工程窗口

工程窗口如图所示，类似 Windows 资源管理器窗口，窗口中列出当前工程中的窗体和模块，以层次化管理方式显示各类文件，而且允许同时打开多个工程。在工程资源管理器标题栏的下方有三个按钮，其含义和作用如下：

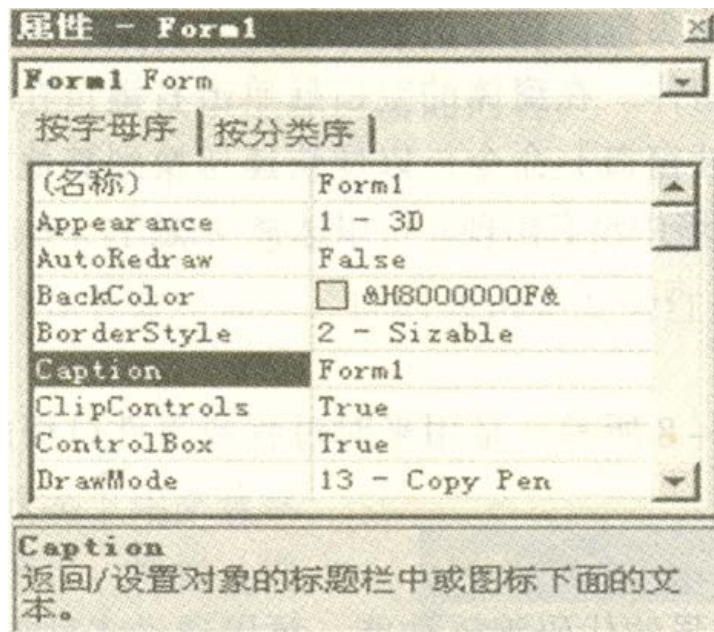
[查看代码]：单击后切换到代码编辑器窗口，查看或编辑代码。

[查看对象]：单击后切换到窗体设计器窗口，查看或设计当前窗体。

[切换文件夹]：单击后可以在工程中的不同层次之间切换。

### 3、属性窗口

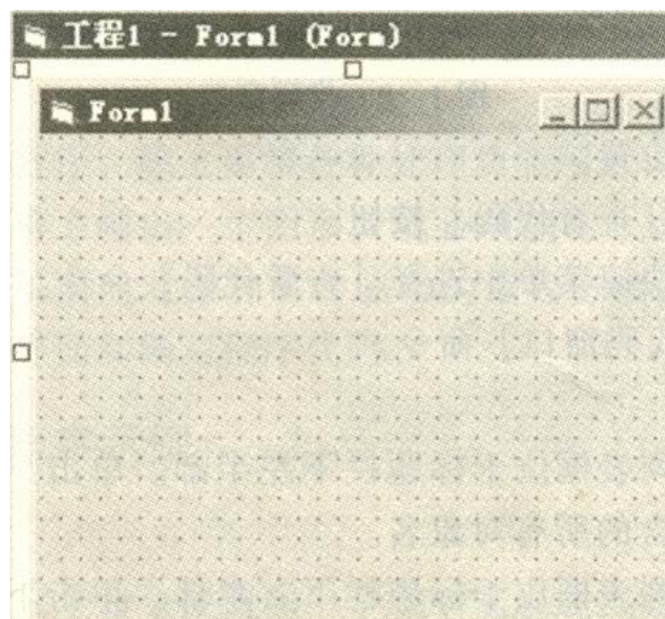
在 VB 集成环境中，属性窗口的默认位置是在工程窗口的下方，如图所示。单击工具栏中的属性窗口按钮或按 F4 键，可以使隐藏起来的属性窗口再次出现。



属性窗口

应用程序中的窗体及其控件的属性,均可以通过属性窗口设置,例如,名称、标识、颜色、字体等。属性窗口由以下几部分组成。

(1)对象下拉列表框:标识当前对象的名称及其所属的类别,例如,下图中 Form 是名称,Form 说明是窗体类。单击属性窗口中对象下拉列表框右边的箭头可列出所选窗体中包含的对象列表。



窗体设计器



(2) 选项卡: 可按字母序或分类序两种方式列出所选对象的所有属性。

(3) 属性列表: 该表中列出所选对象的各个属性的默认值, 可以在设计模式或运行模式中重新设置; 修改其属性值。不同的对象, 列出的属性也不尽相同。列表左边列出的是各种属性, 右边对应的是属性值。

(4) 属性含义: 显示所选取属性的简短文字说明。

#### 4、窗体设计器

"窗体设计器", 是用户设计应用程序界面的窗口, 也称为"对象窗口"。其中的窗体是用来开发 VB 应用程序界面的, 用户可以在窗体中放置各种控件, 窗体中的控件可随意在窗体上移动、放缩, 但控件被锁定后则不可随意修改。

窗体是 VB 应用程序的主要部分, 用户通过与窗体上的控件进行交互得到操作结果。每个窗体必须有一个惟一的窗体名称, 建立窗体时的默认名称为 Form1, Form2, ..., 用户可以根据需要在工程中建立多个窗体。在窗体的空白处单击右键, 在快捷菜单中, 选择[查看代码]、[菜单编辑器]或[属性窗口]命令, 以便快速切换到其他窗口。如果窗体设计器在集成环境中没有出现, 可以选择[视图] → [对象窗口]命令使其再现。

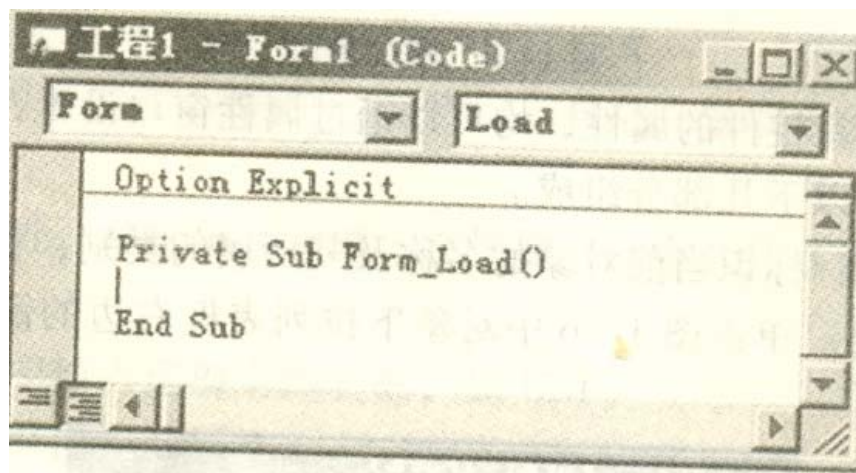
#### 5、代码编辑窗口

"代码编辑窗口"如图 1-8 所示, 是用来对过程或事件过程的代码进行编辑的窗口, 也称为"代码窗口"。

##### (1) 代码窗口的打开

VB 系统为用户提供了较强的代码编辑功能, 可以通过多种方式打开代码编辑窗口。





代码窗口

- ①双击窗体中的任何位置。
- ②单击工程窗口中的[查看代码]按钮。
- ③单击鼠标右键，从快捷菜单中选择[查看代码]命令。
- ④选择[视图] → [代码窗口]命令。

## (2) 代码窗口的组成

①对象下拉列表框。列表框位于标题栏下左半部。单击下拉列表按钮，会弹出列表，列表中给出当前窗体及所包含的所有对象名 0

②过程下拉列表框。列表框位于标题栏下右半部。单击下拉列表按钮，会弹出列表，列表中给出所选对象的所有事件名称。

③代码编辑区。窗口中的空白区域即为代码编辑区。用户可以在其中编辑程序代码，操作方法与通常文字处理软件类似，而且在 VB 代码编辑方面提供了一些自动功能。

④查看视图按钮。在代码窗口的左下角，有"过程查看"和"全模块查看"两个按钮，前者用于查看一个过程，后者可以查看程序中的所有过程。

## (3) 代码编辑器的自动功能

用户在编辑程序代码时可以直接输入语句、函数、对象的属性或方法等内容，也可以利用 VB 提供的自动功能简化输入过程。

①自动提示信息。当用户输入正确的 VB 函数后，在当前行的下面会自动显示出该函数的语法格式，当前项为加黑显示，为用户输入提供参考。输入一项后，下一项又变为加黑显示。

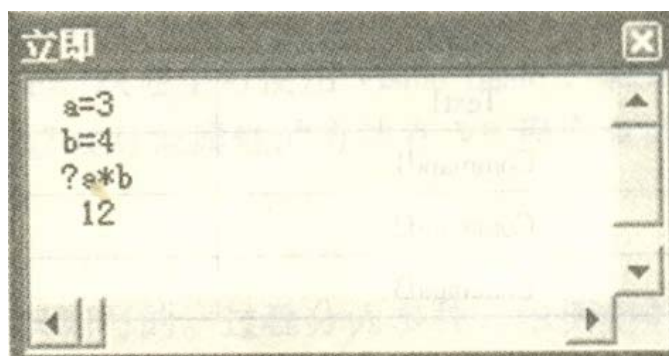
②自动列出成员。用户在输入控件名后面的小数点时，VB 系统会自动弹出下拉列表框，列表中包含了该控件的所有成员：属性、方法，继续输入成员名的字母，系统会自动显示出相关的属性名和方法名，可以从中选择所需的内容。

如果操作中没有出现自动提示信息和自动列出成员的功能，按 Ctrl+J 键可以使其出现。

③自动语法检查。在输入代码的过程中，每次按回车键时，VB 都会自动检查该行语句的语法。如果出现错误，VB 会警告提示，同时该语句变为红色。

## 6、立即窗口

立即窗口是在中断模式状态下用来观察处理结果、调试程序使用的窗口。选择[视图]→[立即窗口]命令，即可打开如图所示的立即窗口。可以在立即窗口中直接输入命令，观察结果；也可以在程序中使用 Debug 对象输出的方式，将结果送到立即窗口。例如，在程序中输入 Debug.printdate 即可在立即窗口中显示系统日期。



立即窗口

## (二) 开发 VisualBasic 应用程序的基本步骤

用传统的高级语言编写程序，除去要将各种处理功能用语句描述之外，还需要设计用户操作界面，设计者为了在屏幕上画出一个图形，就必须编写一大段程序语句。VB 可视化编程改变了这种状况，而采用了面向对象和事件驱动的编程方法。因此用 VB 开发应用程序，包括两部分工作：设计用户界面和编写程序代码。

### 1、VisualBasic 应用程序示例

#### (1) 题目要求和运行效果

设计一个程序，界面由三个命令按钮和一个文本框组成，如图所示。当用户单击[欢迎]按钮时，在文本框中显示"欢迎学习使用 VisualBasic"；单击[清除]按钮时，清除文本框的内容；单击[退出]按钮时，退出程序的运行。



应用程序实例窗体

## (2) 操作步骤

①新建工程。选择[文件]→[新建工程]命令，在出现的"新建工程"对话框中选择[标准 EXE]，然后单击[确定]按钮。

②添加文本框控件。把鼠标指针移到工具箱中的文本框图标 ab1 上，单击鼠标，鼠标指针变形为"+", 然后将鼠标移到窗体合适的位置上，按住鼠标左键做斜线拖动，当控件的尺寸符合需要时，释放鼠标，文本框被添加到窗体中。

③添加按钮控件。添加按钮与添加文本框的操作相似，区别仅仅是在工具箱中选择的工具不同。连续添加三个按钮控件。

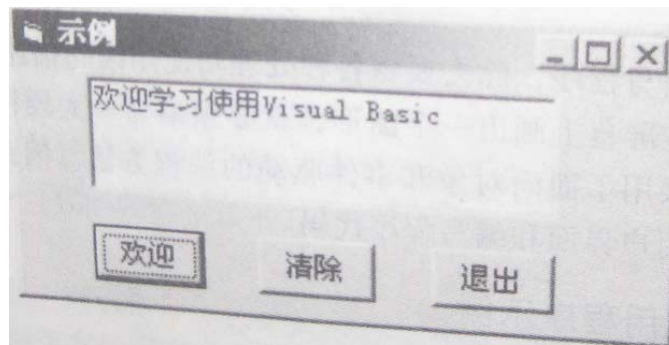
④设置控件属性。按照下表所示设置窗体、文本框和命令按钮的属性。

示例中的对象属性设置

控件名称	属性	属性值
Form1	Caption	示例
Text1	Text	一
Command1	Caption	欢迎
Command2	Caption	清除
Command3	Caption	退出

⑤编写代码。在工程窗口单击[查看代码]按钮，在对象列表框中分别选择三个按钮对应的名称，出现三个事件过程的框架，在其中输入相应的代码即可。

⑥运行程序。当确认代码正确无误时，单击[启动]按钮。运行后的界面如图所示。



示例运行界面

### 3. 程序代码

```
PrivateSubCommand1_Click()
```

```
    Text1.Text="欢迎学习使用 VisualBasic"
```

```
EndSub
```

```
PrivateSubCommand2_Click()
```

```
    Text1.Text=""
```

```
EndSub
```

```
PrivateSubCommand3_Click()
```

```
    End
```

```
EndSub
```

## 2、基本步骤

通过以上对程序示例处理过程的描述，可以归纳出开发 VB 应用程序的如下基本步骤。

### (1) 界面设计

应用程序界面由窗体和控件两部分组成，用户可以在窗体中放置各种控件，以构成用户所需要的程序界面。在窗体上设计界面时，可以使用前面介绍的鼠标拖动的方法，也可以用鼠标双击工具箱中的控件按钮，这样

窗体的中心位置会自动出现相应的控件，然后再将控件移动到合适的位置，调整其尺寸大小。可以使用格式菜单中的命令使控件对齐、调整尺寸、设置间距。为了防止处于理想位置的控件被无意移动，可以选择格式菜单中的锁定控件命令。如果需要对个别控件的位置做微调，按住 Ctrl 键后，再用方向箭头来调整。

## （2）属性设置

在介绍属性窗口和程序示例的过程中，对有关属性设置的基本问题已经作了初步介绍。可以在属性窗口中静态设置对象的属性，也可以在程序中对属性进行设置或修改，进行动态的设置。例如，前面示例程序中的语句 `Text.Text="欢迎学习使用 VisualBasic"`，就是在程序运行阶段，修改了文本框的文本属性。这种动态修改对象属性的方法在 VB 程序设计的过程中经常用到。

## （3）编写程序代码

在 VB 中，程序代码是集中在"过程"中组织编写的。过程分为多种，示例中的程序代码是在事件过程中编写的。事件过程中的代码是为响应在一个对象上发生的"事件"所进行搞操作。例如，示例中的 `Command2_Click()` 事件过程，将对象和单击事件结合起来构成一个事件过程。当单击"清除"按钮的事件发生时，执行了 `Text.Text=""` 的操作命令，体现了 VB 事件驱动的机制。

## （4）运行和保存程序

可以通过多种方式运行一个 VB 应用程序。

①选择[运行]→[启动]命令。

②单击工具栏中的[启动]按钮。

③按 F5 功能键。

可以通过多种方式结束正在运行的 VB 应用程序。

①选择[运行]→[结束]命令。

②单击工具栏中的[结束]按钮。

保存 VB 应用程序的操作与其他 Windows 应用程序类似。但是，在 VB 的一个工程中包含多个不同类型的文件，这些文件需要分别保存。选择[文件]→[保存 Form1]命令，即可保存第一个窗体；选择[文件]→[保存工程]命令，即可保存当前的工程文件。

#### (5) 独立运行 VB 应用程序

VB 应用程序可以脱离 VB 的集成环境，直接在 Window 状态下运行。因此需要对应用程序进行编译，生成 .exe 文件。选择[文件]→[生成工程 1.exe]命令，在出现的对话框中，输入可执行文件的名称，单击[确定]按钮，即可生成在 Window 中直接运行的文件。

## AE 实验预备知识

ArcGIS Engine 9.2 提供了一些功能非常强大的控件可以帮助开发人员快速开发自己的 GIS 应用，例如：

MapControl

PageLayoutControl

SceneControl

GlobeControl

ToolbarControl



TOControl

SymbologyControl (9.2 版本新增控件)

LicenseControl

## 实验二 地图浏览

### 一、实验目的

通过操作，理解 ArcGIS Engine 的特点和用法

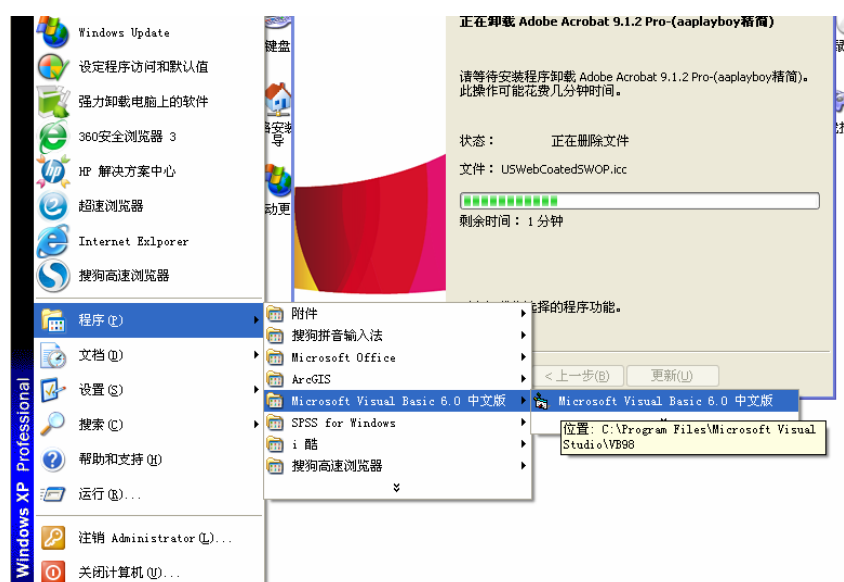
### 二、实验任务

本示例在不写任何代码的情况下，创建一个地图浏览小程序，可以打开 mxd 地图文档，对地图进行缩放、漫游、点击查询属性等。

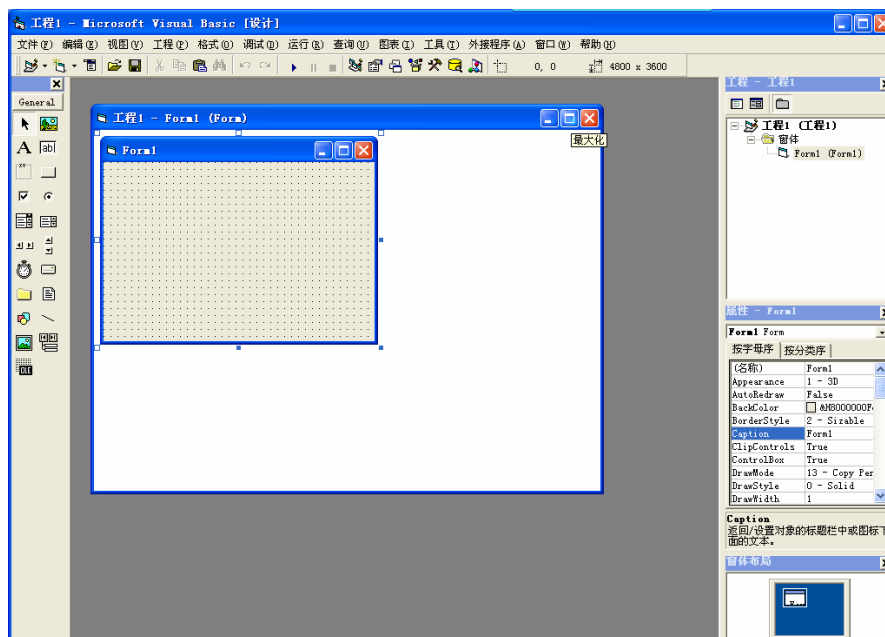
### 三、实验内容

#### (一) 启动 VisualBasic6.0

1、从开始菜单或快捷方式启动 VisualBasic6.0，启动画面如下。



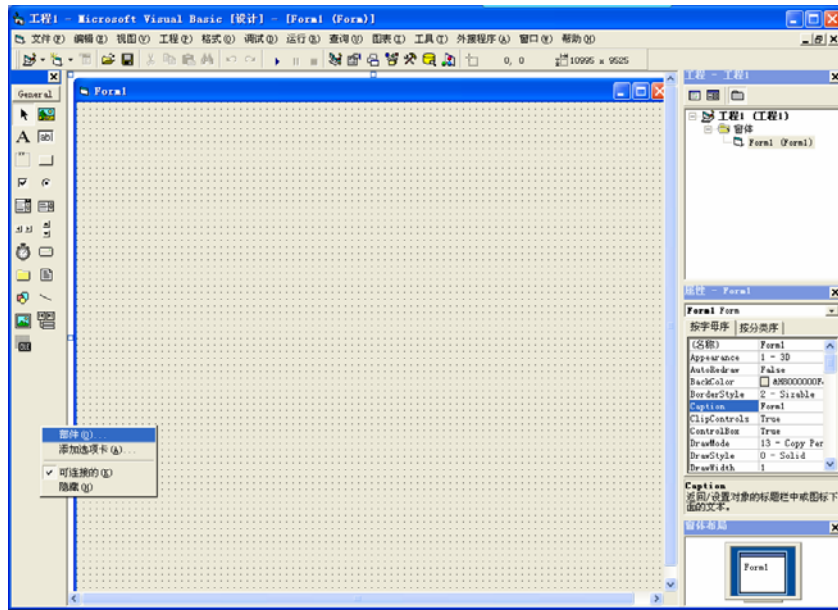
2、在 VisualBasic 启动画面弹出的新建工程窗口选择“新建”选项卡，选择“标准 EXE”，然后点击打开，即创建了一个新的窗体。



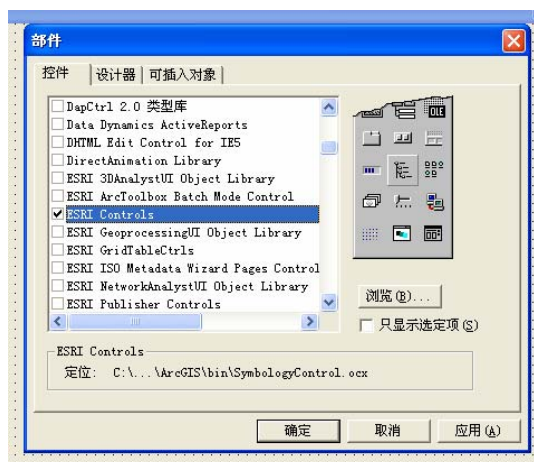
3、设置新创建窗体的大小。点击窗体上面最大化符号，将窗体最大化

## (二) 加载 ArcGIS Engine 控件

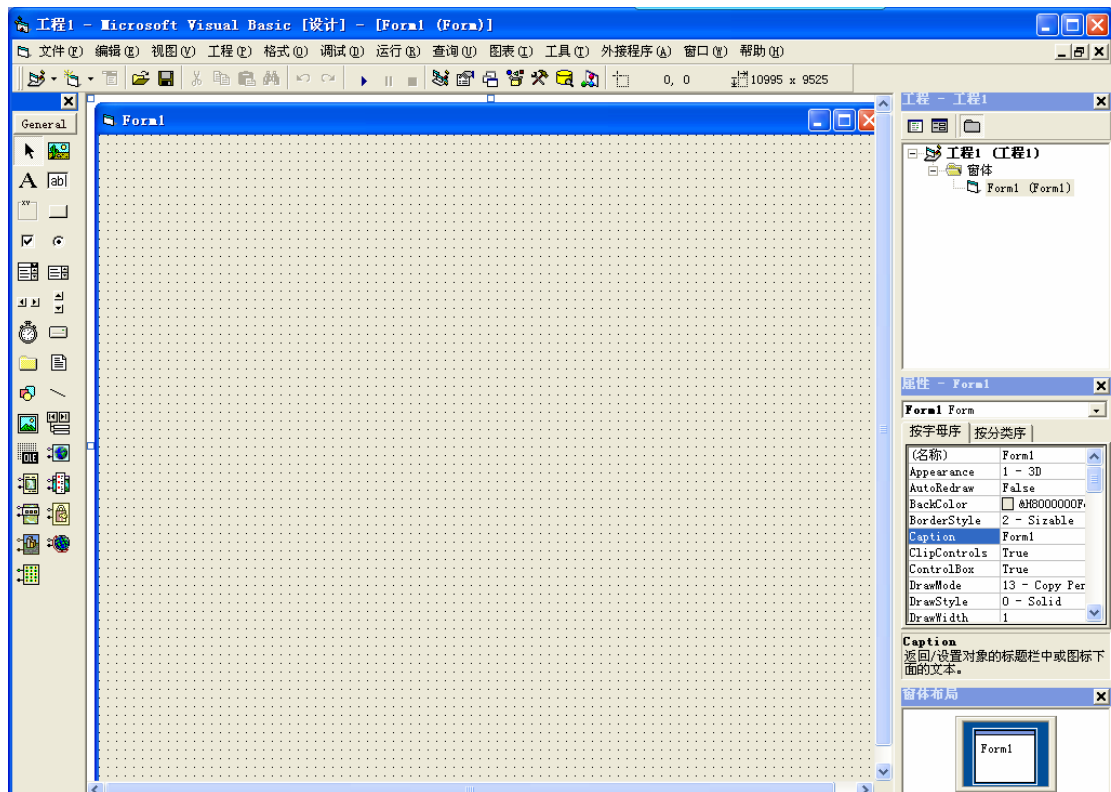
1、在 VisualBasic 界面左侧控件条部分点右键选择“部件”，如下图



2、在弹出的对话框中，选择加载“ESRIControls”，点击“确定”。



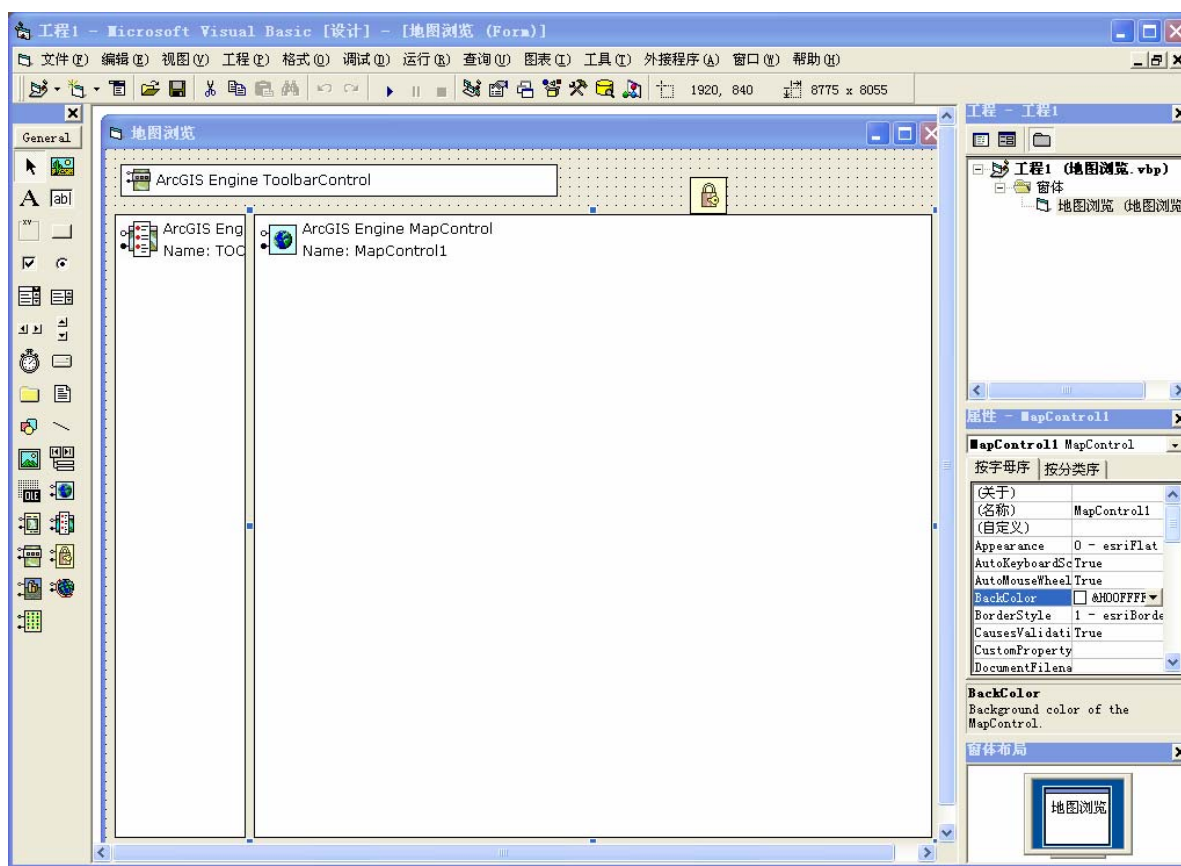
选择加载“ESRIControls”控件



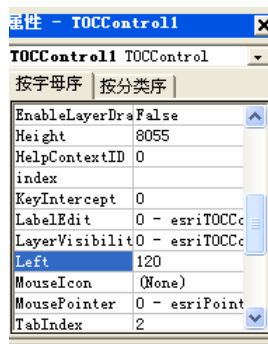
加载“ESRIControls”控件后

(三) 在 Form1 窗体上使用 MapControl、ToolBarControl、TOCControl、LicenseControl 控件

1、在 Form1 窗体界面上使用鼠标拖动以上四个控件，使各个控件如下图所示，窗体顶部是工具栏 (ToolBar)，左侧是图层列表 (TOC)，主工作区是地图控件 (Map)。

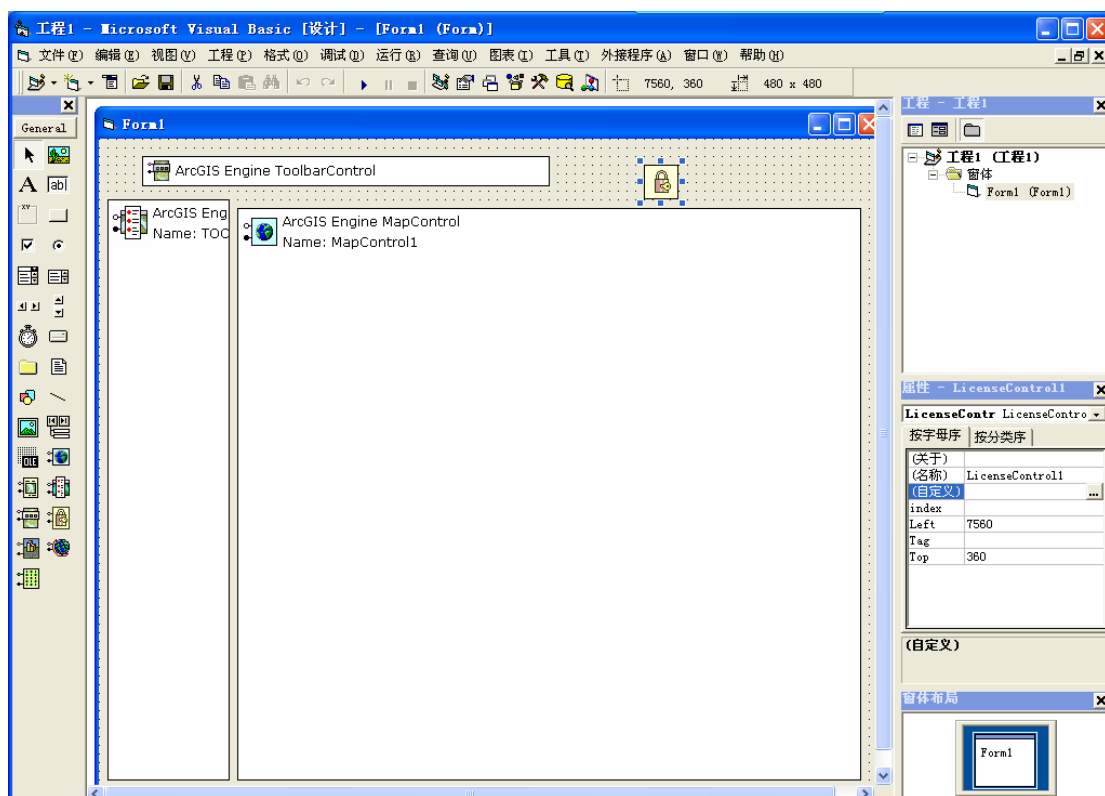


2、通过调整各个控件属性中的 Left、Top、Height、Width 等设置各个控件的位置和形状等；

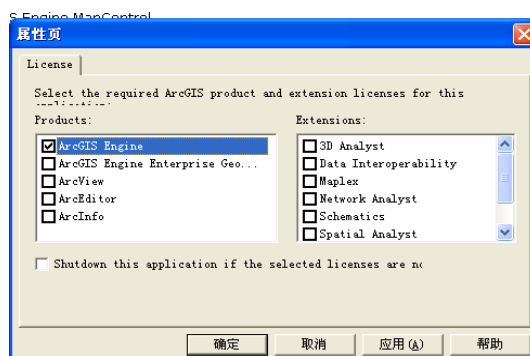


#### (四) 使用 LicenseControl 控件添加要使用的模块功能

1、选中窗体上的 LicenseControl 控件，在右侧属性项中点击“自定义”，

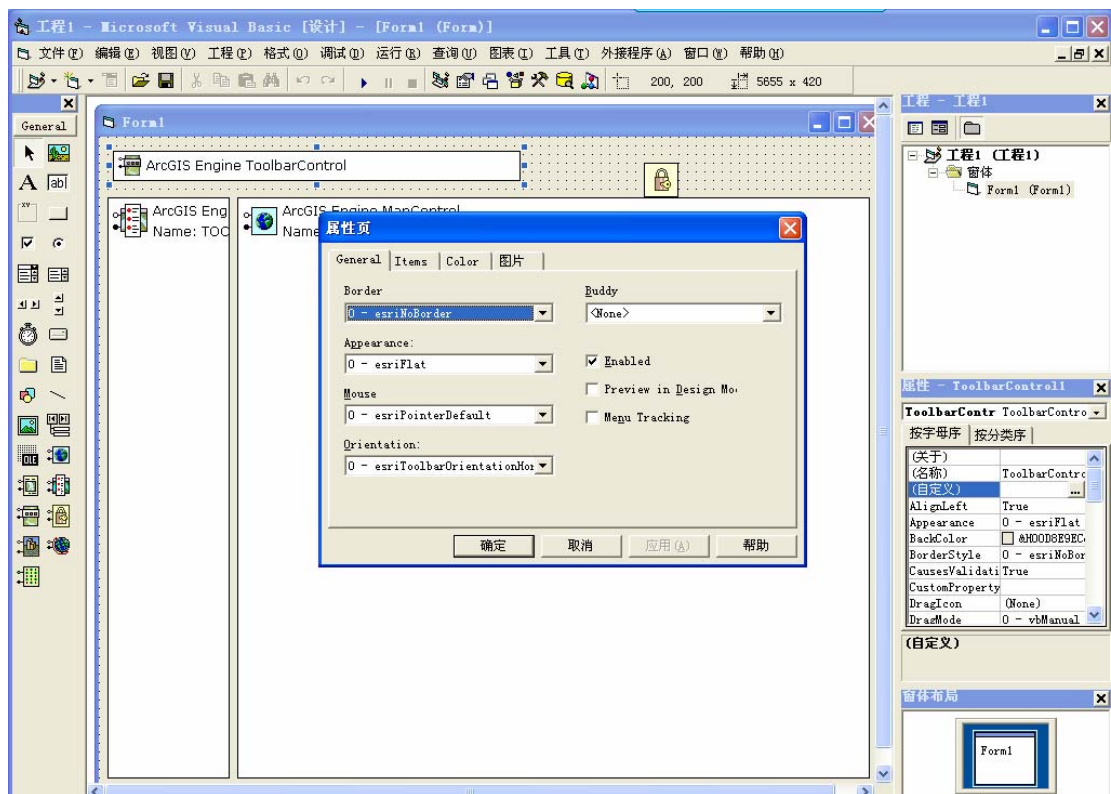


2、在弹出对话框中，选中要加载的功能模块，默然加载 ArcGIS Engine 模块许可。

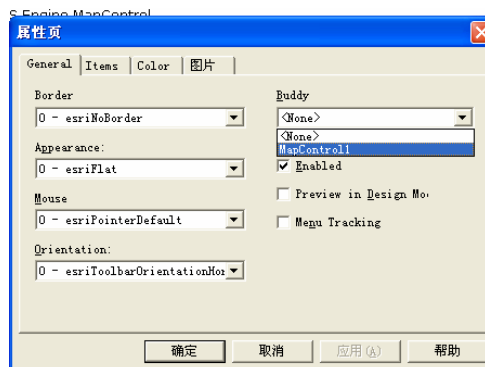


## (五) 使用 ToolbarControl1 控件加载工具

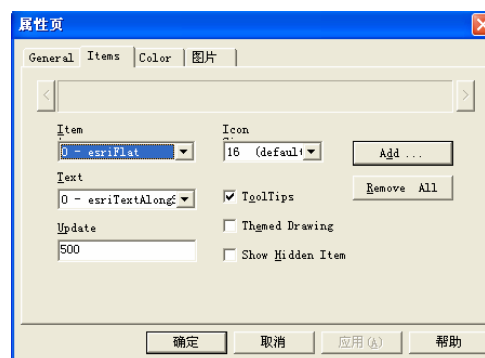
1、选中窗体上的 ToolbarControl1 控件，在右侧属性项中点击“自定义”，弹出如下对话框



2、在弹出的对话框中，首先设置 General-Buddy-MapControl1

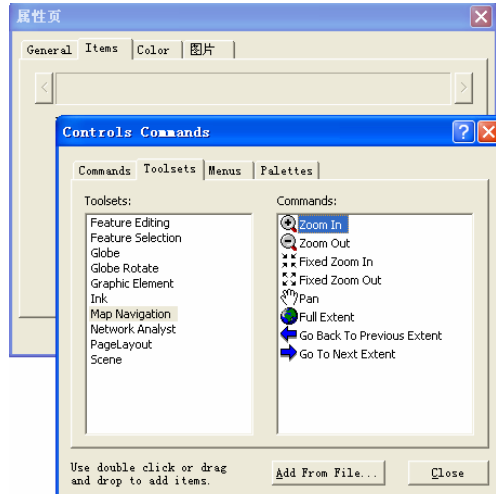


3、选择 Items 选项，点击 Add 按钮，添加工具

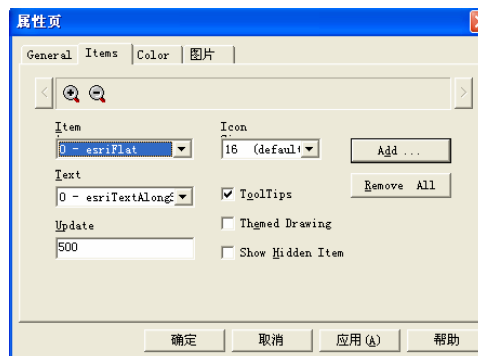


4、在弹出的对话框中，选择 Toolsets-MapNavigation，双击 ZoomIn或ZoomOut 等工具。



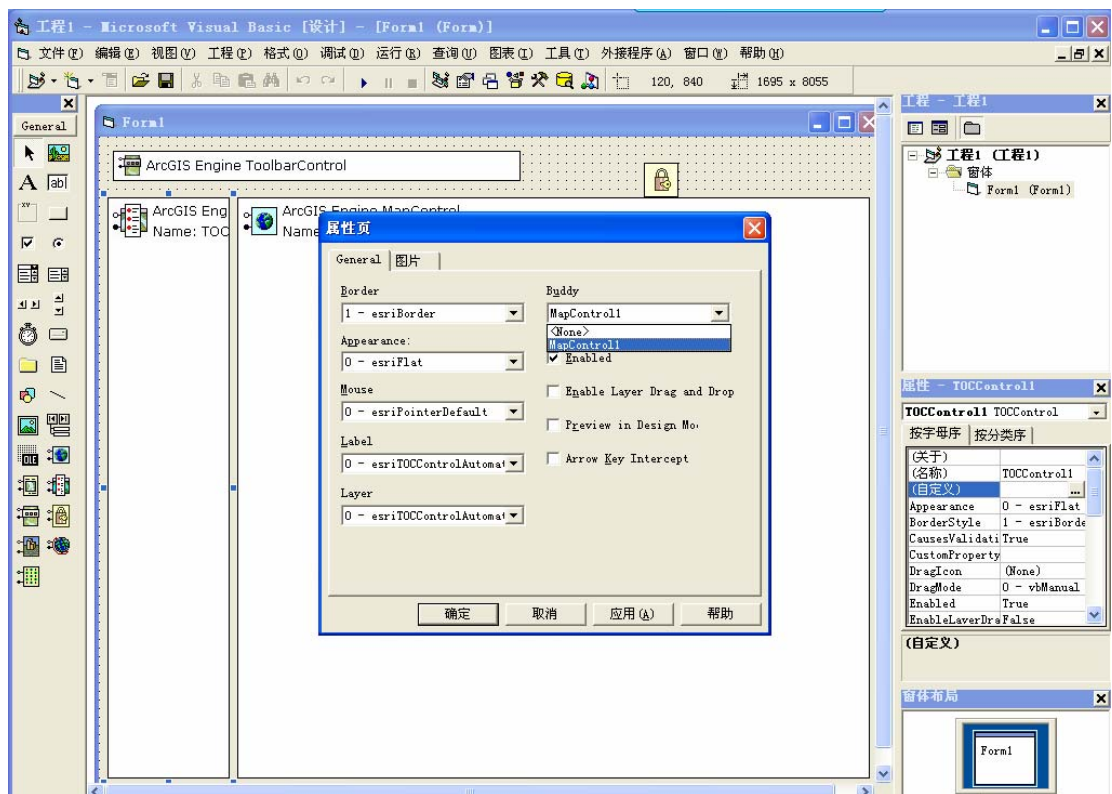


5, 点击确定, 将选中的工具添加到 ToolbarControl 控件、

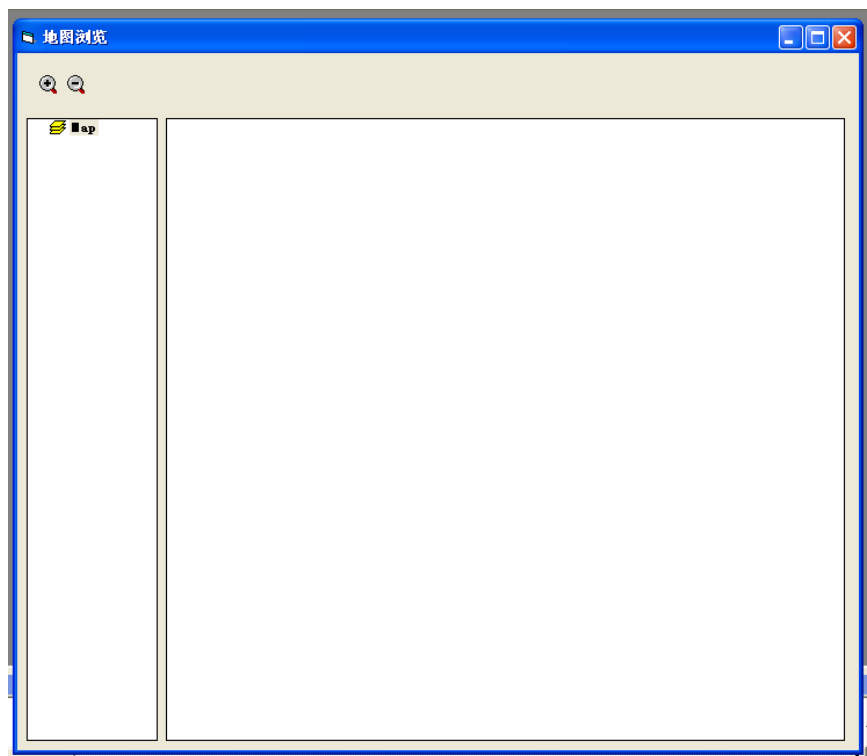


## (六) 使用 TOCControl 控件

1、选中窗体上的 ToolbarControl 控件, 在右侧属性项中点击“自定义”, 弹出如下对话框, 并设置 General-Buddy-MapControl1



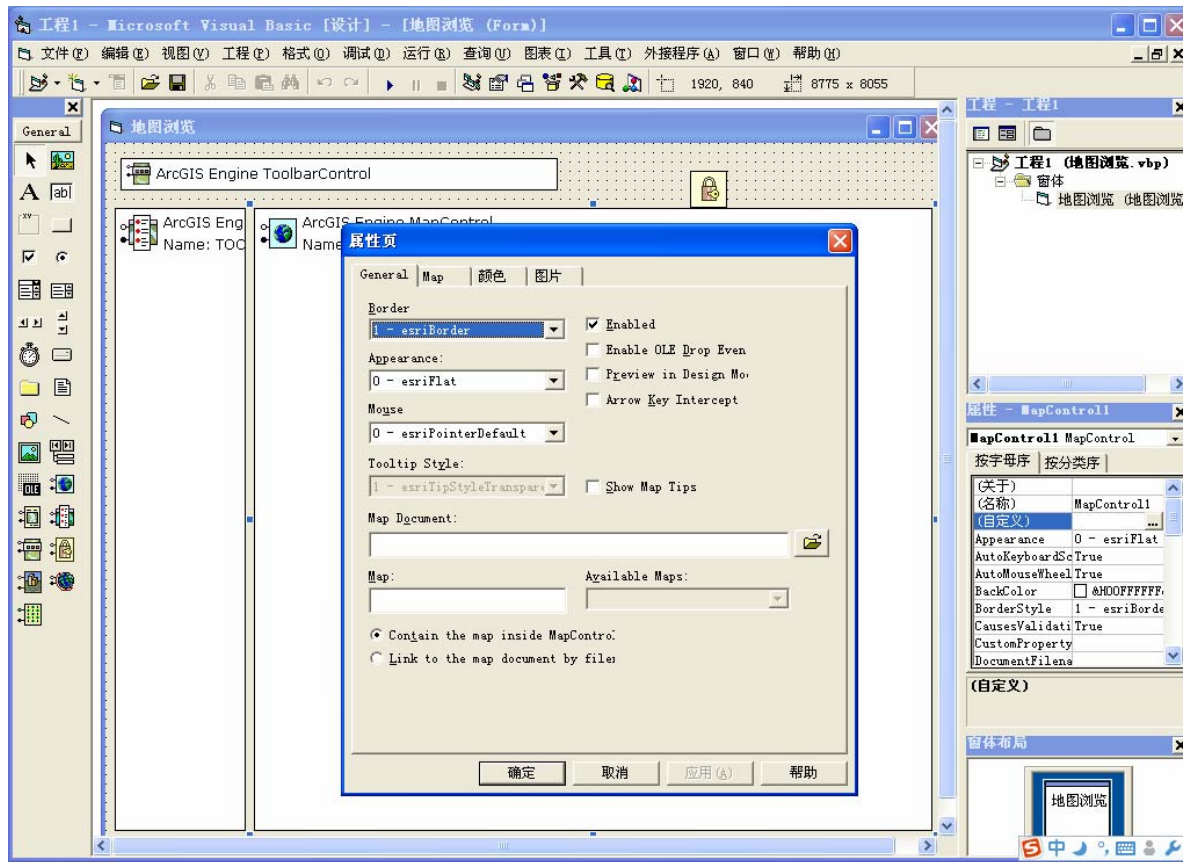
2、运行，查看结果。选择菜单-运行，即可查看运行结果



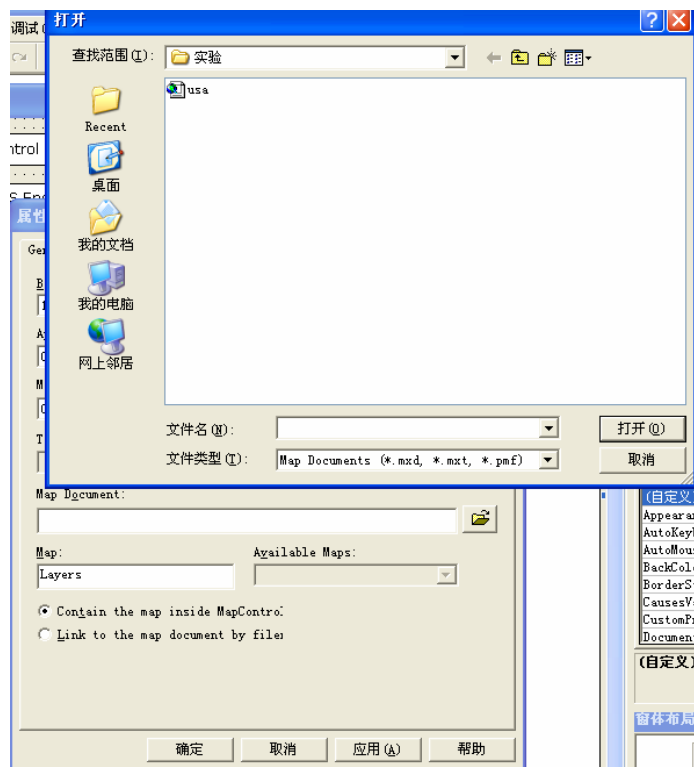
(七) 使用 MapControl 控件加载数据

1、选中窗体上的 ToolbarControl 控件，在右侧属性项中点击“自定

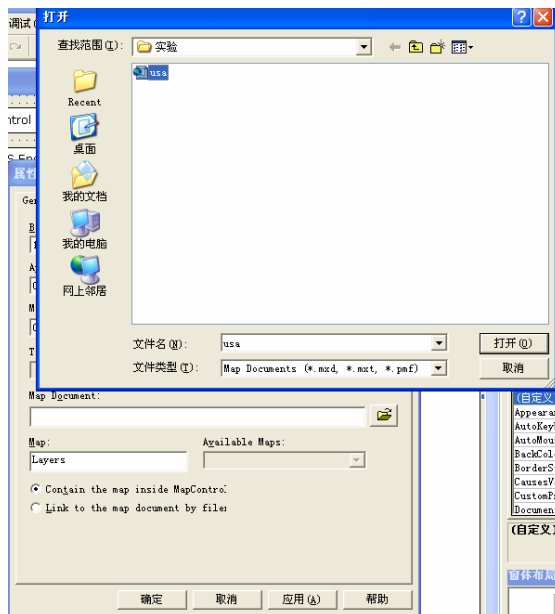
义”，弹出如下对话框，并设置 General1-Buddy-MapControl1



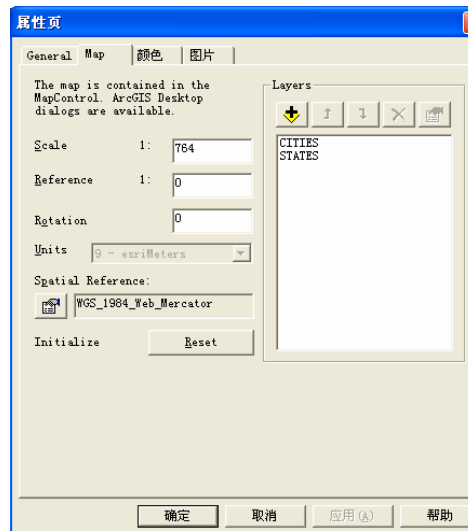
2、在 MapDocument 右侧，选择打开文档按钮，弹出如下对话框



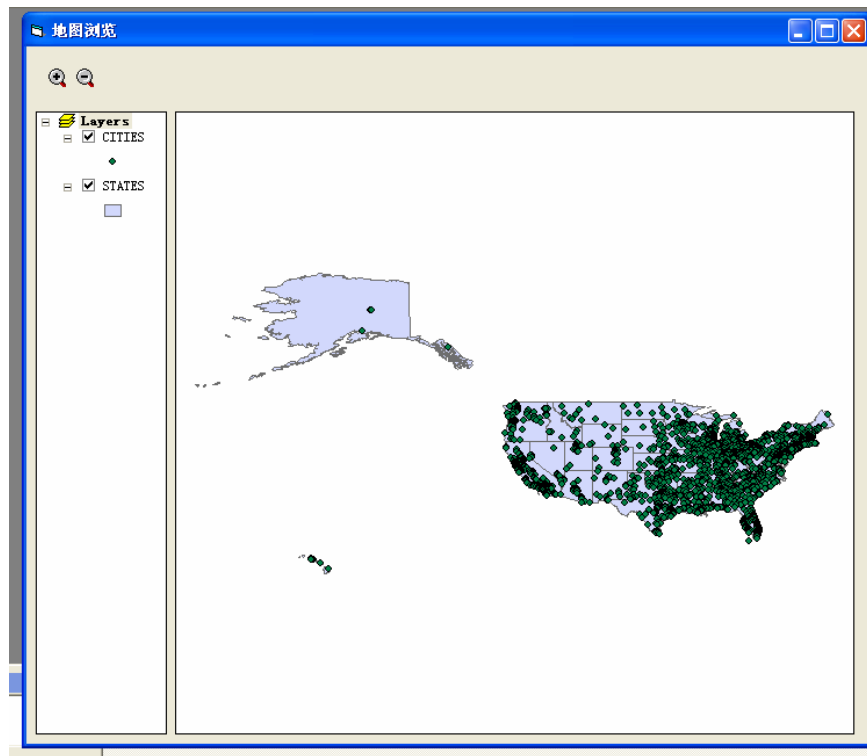
### 3、选择打开实验数据 “usa.mxd” 文档



### 4、在 map 选项可以查看该文档内的图层及坐标参数



### 5、点击确定，并运行



#### 四、练习题

（一）练习加载打开文档、数据漫游、数据查询、添加数据等工具；

（二）利用（一）题中的打开文档工具，打开 usa.mxd 文档；

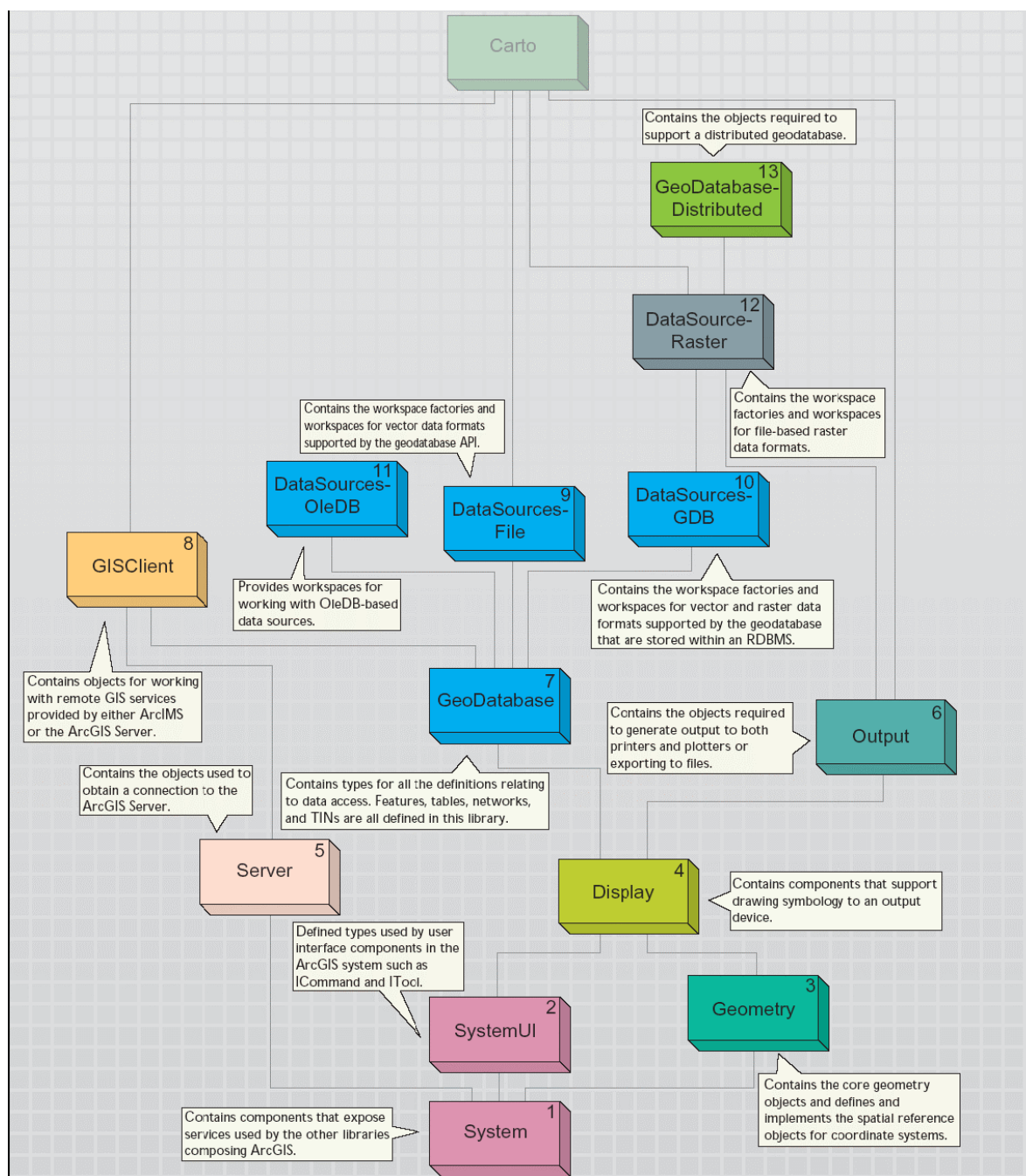
（三）利用（一）题中的数据查询工具，查询 usa.mxd 文档中 States 图层的属性。

### 实验三 ArcGIS Engine 的类库

本实验概述 ArcGIS Engine 中的各个类库。各个模型图展示了 ArcGIS Engine 的类库架构。理解类库结构、它们的依赖关系和基本功能将有助于开发人员了解 ArcGIS Engine 的组件。

下面根据依赖关系的顺序对类库进行讨论。模型图中在每个类库框的右上角显示了其序列号。例如，作为 ArcGIS 体系结构基础的 System 类库，其编号为 1，而编号为 7 的 GeoDatabase 类库依赖于模型图中其前面的 6 个类库—SyStem、SystemUI、Geometry、Display、Server 和 Output。重

点掌握前 6 个类库，即名称下有下划线的类库。



## 一、System 类库

System 类库是 ArcGIS 体系结构中最底层的类库。System 类库包含给构成 ArcGIS 的其他类库提供服务的组件。System 类库中定义了大量开发者可以实现的接口。AoInitializer 对象就是在 System 类库中定义的，所有的开发者必须使用这个对象来初始化 ArcGIS Engine 和解除 ArcGIS

Engine 的初始化。开发者不能扩展这个类库，但可以通过实现这个类库中包含的接口来扩展 ArcGIS 系统。

## 二、SystemUI 类库

SystemUI 类库包含用户界面组件接口定义，这些用户界面组件可以在 ArcGIS Engine 中进行扩展。包含 ICommand、ITool 和 IToolControl 接口。开发者用这些接口来扩展 UI 组件，ArcGIS Engine 开发人员自己的组件将使用这些 UI 组件。这个类库中包含的对象是一些实用工具对象，开发人员可以通过使用这些对象简化用户界面的开发。开发者不能扩展这个类库，但可以通过实现这个类库中包含的接口来扩展 ArcGIS 系统。

## 三、Geometry 类库

Geometry 类库处理存储在要素类中的要素几何图形或形状或其他图形元素。大多数用户会遇到的基本几何图形对象为 Point、MultiPoint、Polyline 和 Polygon。除了这些顶级实体外，就是作为多义线和多边形的组成部分的几何图形，是组成几何图形的子要素。它们是 Segment、Path 和 Ring。Polyline 和 Polygon 由一系列相连接的、构成 Path 的片段组成。一个片段由两个不同的点，即起始点和结束点，以及一个定义这两点之间弯曲度的元素类型组成。片段的类型有 CircularArc、Line、EllipticArc 和 BezierCurve。所有几何图形对象都可以有与其顶点相关联的 Z、M 和 IDs。所有的基本几何图形对象都支持诸如 Buffer、Clip 等几何操作。几何子要素不可以由开发者扩展。

GIS 中的实体指的是现实世界中的要素；现实世界中要素的位置由一个带有空间参考的几何图形来定义。空间参考对象，包括投影坐标和地理



坐标系统，都包括在 Geometry 类库中。开发者可以通过添加新的空间参考和投影来扩展空间参考系统。

#### **四、Display 类库**

Display 类库包含用于显示 GIS 数据的对象。除了负责实际输出图像的主要显示对象外，这个类库还包含表示符号和颜色的对象，它们用来控制在显示上绘制时实体的属性。Display 类库还包含在与显示交互时提供给用户可视化反馈的对象。开发者与 Display 最常用的交互方式就是类似于 Map 对象或 PageLayout 对象提供的视图。Display 类库的所有部分都能进行扩展；通常扩展的对象包括符号、颜色和显示反馈。

#### **五、Server 类库**

Server 类库包含允许用户连接并操作 ArcGIS Server 的对象。开发人员用 GIS Server Connection 对象来访问 ArcGIS Server。通过 GIS Server Connection 可以访问 Server Objects Manager 对象。用这个对象，开发人员可以操作 Server Context 对象，以处理运行于服务器上的 ArcObjects。开发人员还可以用 GIS Client 类库与 ArcGIS Server 进行交互。

#### **六、Output 类库**

Output 类库用于创建图形输出到诸如打印机和绘图仪等设备以及诸如增强型元文件和栅格图像格式（JPG、BMP 等）等硬拷贝格式。开发人员用这个类库中的对象及 ArcGIS 系统的其他部分创建图形输出。通常是 Display 和 Carto 类库中的对象。开发者可以为自定义设备和输出格式扩展 Output 类库。

## 七、GeoDatabase 类库

Geodatabase 类库为地理数据库提供了编程 API。地理数据库是建立在标准工业关系型和对象关系数据库技术之上的地理数据仓库。Geodatabase 类库中的对象为 ArcGIS 支持的所有数据源提供了一个统一编程模型。GeoDatabase 类库定义了许多由 ArcObjects 架构中更高级的数据源提供者实现的接口。开发者可以扩展地理数据库,以支持特定类型的数据对象(要素、类等);此外,Geodatabase 类库还有用 PlugInDataSource 对象添加的自定义矢量数据源。地理数据库支持的本地数据类型不能扩展。

## 八、GISClient 类库

GISClient 类库允许开发者使用 Web 服务;这些 Web 服务可以由 ArcIMS 和 ArcGIS Server 提供。GISClient 类库中包含用于连接 GIS 服务器以栅格数据对象(RDO)是一个 COMAPI,提供支持基于文件的栅格数据的显示和分析功能。

使用 Web 服务的对象。该类库支持 ArcIMS 的图像和要素服务。GISClient 类库提供以无态方式直接或通过 Web 服务目录操作 ArcGISServer 对象的通用编程模型。在 ArcGISServer 上运行的 ArcObjects 组件不能通过 GISClient 接口来访问。要直接获得访问在服务器上运行的 ArcObjects,开发人员应使用 Server 类库中的功能。

## 九、DataSourcesFile 类库

DataSourcesFile 类库包含用于基于文件数据源的 GeoDatabaseAPI 实现。这些基于文件的数据源包括 shapefile、coverage、TIN、CAD、SDC、

StreetMap 和 VPF。开发者不能扩展 DtaSourcesFile 类库。

## **十、DataSourcesGDB 类库**

DataSourcesGDB 类库包含用于数据库数据源的 GeoDatabaseAPI 实现。这些数据源包括 MicrosoftAccess 和 ArcSDE 支持的关系型数据库管理系统——IBM、DB2、Informix、MicrosoftSQLServer 和 Oracle。开发者不能扩展 DataSourcesGDB 类库。

## **十一、DataSourcesOLEDB 类库**

DataSourcesOLEDB 类库包含用于 MicrosoftOLEDB 数据源的 GeoDatabaseAPI 实现。此类库只能用在 MicrosoftWindows 操作系统上。这些数据源包括支持数据提供者和文本文件工作空间的所有 OLEDB。开发者不能扩展 DataSourcesOLEDB 类库。

## **十二、DataSourcesRaster 类库**

DataSourcesRaste 类库包含用于栅格数据源的 GeoDatabaseAPI 实现。这些数据源包括 ArcSDE 支持的关系型数据库管理系统——IBM、DB2、Informix、MicrosoftSQLServer 和 Oracle，以及其支持的 RDO 栅格文件格式。当需要支持新的栅格格式时，开发者不扩展这个类库，而是扩展 RDO。开发者不能扩展 DataSourcesRaster 类库。

## **十三、GeoDatabaseDistributed 类库**

GeoDatabaseDistributed 类库通过提供地理数据库数据导入和导出工具，可以支持对企业级地理数据库的分布式访问。开发者不能扩展 GeoDatabaseDistribute 类库。

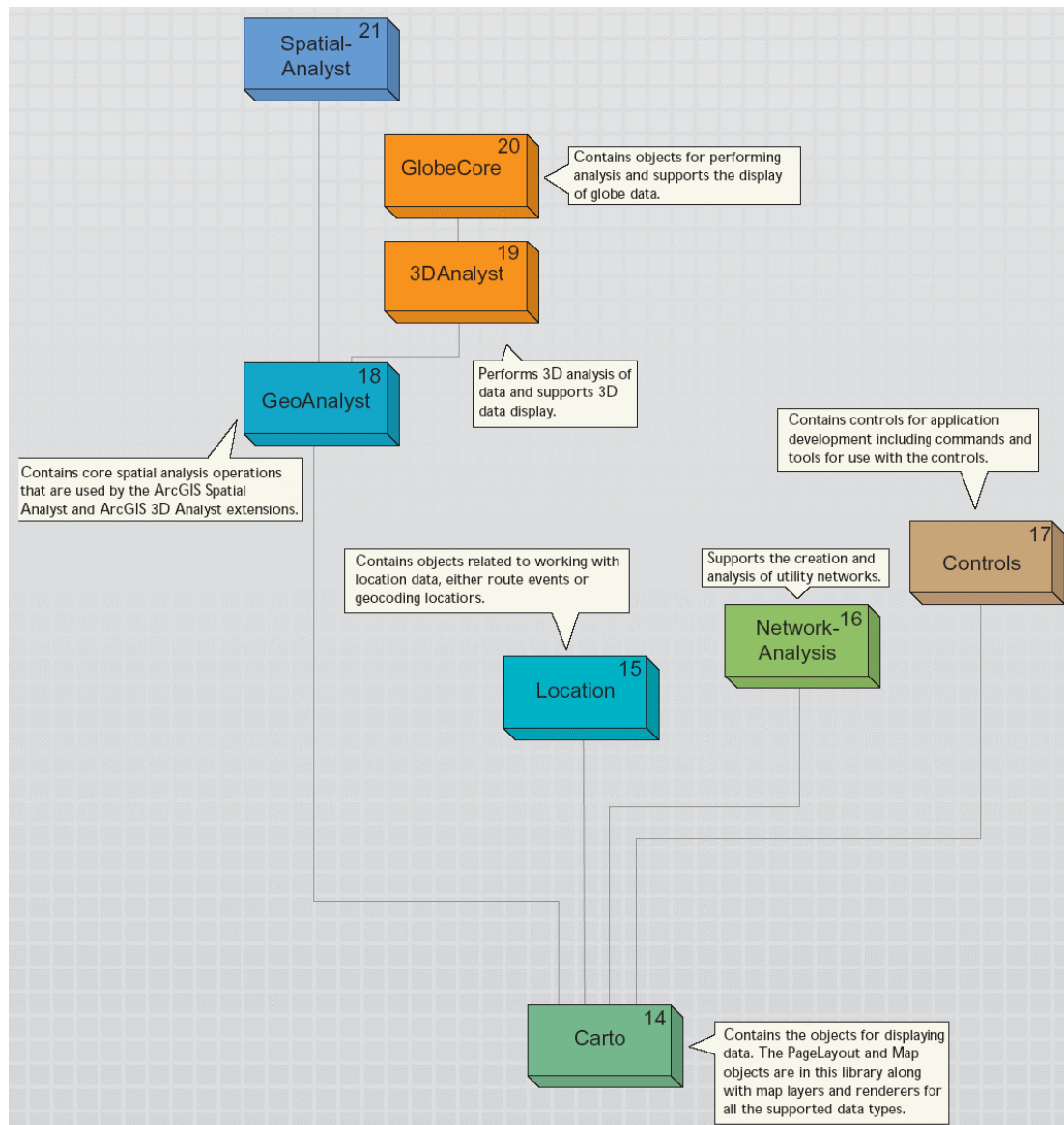
## **十四、Carto 类库**

Carto 类库支持地图的创建和显示；这些地图可以在一幅地图或由许多地图及其地图元素组成的页面中包含数据。PageLayout 对象是驻留一幅或多幅地图及其底土元素的容器。地图元素包括指北针、图例、比例尺等。Map 对象包括地图上所有图层都有的属性——空间参考、地图比例尺等，以及操作地图图层的方法。可以将许多不同类型的图层加载到地图中。

不同的数据源通常有相应的图层负责数据在地图上的显示，矢量要素由 FeatureLayer 对象处理，栅格数据由 RasterLayer 对象处理，TIN 数据由 TinLayer 对象处理，等等。必要的话，图层可以处理与之相关数据的所有绘图操作，但通常图层都是一个相关的 Renderer 对象。Renderer 对象的属性控制着数据在地图中的显示方式。Renderers 通常用 Display 类库中的符号来进行实际绘制，而 Renderer 只是将特定符号与待绘实体的属性相匹配。Map 对象和 PageLayout 对象可以包含元素。元素用其几何图形定义其在地图或页面上的位置，用行为控制元素的显示。包括用于基本形状、文字标注和复杂标注等的元素。Carto 类库还支持地图注释和动态标注。

尽管开发者可以在其应用程序中直接使用 Map 和 PageLayout 对象，但通常来说开发者更经常使用更高级的对象，如 MapControl、PageLayoutControl 或 ArcGIS 应用程序。这些高级对象简化了一些任务，尽管它们也提供对更低级别的 Map 和 PageLayout 对象的访问，允许开发者更好的控制对象。Map 和 PageLayout 对象并不是 Carto 类库中提供地图和页面绘制的仅有对象。MxdServer 和 MapServer 对象都支持地图和页面的绘制，但不是绘制到窗口中，而是直接绘制到文件中。开发者可以用

MapDocument 对象保存地图和地图文档 (.mxd) 中页面布局的状态, 以便在 ArcMap 或 ArcGIS 控件中使用。Carto 类库通常可以在许多方面进行扩展。自定义 Renderer、Layer 等都很普遍。自定义图层通常是向地图应用程序中加载自定义数据最简单的方法。



## 十五、Location 类库

Location 类库包含支持地理编码和操作路径事件的对象。地理编码功能可以通过细粒度对象来完全控制访问, 或通过 GeocodeServer 对象提供的简化 API 来访问。开发者可以创建自己的地理编码对象。线性参考功能

提供对象用于向线性要素添加事件，用各种绘制方法来绘制这些事件。开发者可以扩展线性参考功能。

## 十六、NetworkAnalysis 类库

NetworkAnalysis 类库提供用于在地理数据库中加载网络数据的对象并提供对象用于分析加载到地理数据库中的网络。开发者可以扩展 NetworkAnalysis 类库以便支持自定义网络追踪。这个类库目的在于操作公共网络：供气管线、电力供应线网等。

## 十七、Controls 类库

开发者用 Controls 类库来构建或扩展具有 ArcGIS 功能的应用程序。ArcGISControls 通过封装 ArcObjects 并提供粗粒度的 API 简化了开发过程。尽管这些控件封装了细粒度的 ArcObjects，但是并不限制对这些细粒度的 ArcObjects 的访问。MapControl 和 PageLayoutControl 分别封装了 Carto 类库的 Map 和 PageLayout 对象。ReaderControl 同时封装了 Map 和 PageLayout 对象，且在操作控件时提供了简化的 API。如果授权了地图发布程序，开发者可以以访问 Map 和 PageLayout 控件类似的方式访问内部对象。Controls 类库还包含实现一个目录表的 TOCControl 及驻留操作合适控件的命令和工具的 ToolbarControl。

开发者通过创建自己的用于操作控件的命令和工具来扩展 Controls 类库。为此 Controls 类库提供 HookHelper 对象。这个对象使得创建一个操作任何控件及操作诸如 ArcMap 这样的 ArcGIS 应用程序的命令变得非常简单。

## 十八、GeoAnalyst 类库

GeoAnalyst 类库包含支持核心空间分析功能的对象。这些功能用在 SpatialAnalyst 和 3DAnalyst 两个类库中。开发者可以通过创建新类型的栅格操作来扩展 GeoAnalyst 类库。为使用这个类库中的对象，需要 ArcGIS SpatialAnalyst 或 3DAnalyst 扩展模块许可，或者 ArcGIS Engine 运行时空间分析或 3D 分析选项许可。

## 十九、3DAnalyst 类库

3DAnalyst 类库包含操作 3D 场景的对象，其方式与 Carto 类库包含操作 2D 地图的对象类似。Scene 对象是 3DAnalyst 类库中主要对象之一，因为该对象与 map 对象一样，是数据的容器。Camera 和 Target 对象规定在考虑要素位置与观察者关系时场景如何浏览。一个场景由一个和多个图层组成；这些图层规定了场景中包含的数据及这些数据如何显示。开发者很少扩展 3DAnalyst 类库。为使用这个类库中的对象，需要 ArcGIS 3DAnalyst 扩展模块许可或 ArcGIS Engine 运行时 3D 分析选项许可。

## 二十、GlobeCore 类库

GlobeCore 类库包含操作 globe 数据的对象，其方式与 Carto 类库包含操作 2D 地图的对象类似。Globe 对象是 GlobeCore 类库中主要对象之一，因为该对象与 Map 对象一样，是数据的容器。GlobeCamera 对象规定在考虑 globe 位置与观察者关系时 globe 如何浏览。一个 globe 有一个和多个图层；这些图层规定了 globe 中包含的数据及这些数据如何显示。

GlobeCore 类库中有一个开发控件及与其一起使用的命令和工具。该开发控件可以与 Controls 类库中的对象协同使用。

开发者很少扩展 GlobeCore 类库。为使用这个类库中的对象，需要

ArcGIS3DAnalyst 扩展模块许可或 ArcGIS Engine 运行时 3D 分析选项许可。

## 二十一、SpatialAnalyst 类库

SpatialAnalyst 类库包含在栅格数据和矢量数据上执行空间分析的对象。开发者通常使用这个类库中的对象，而不扩展这个类库。为使用这个类库中的对象，需要 ArcGIS 空间分析扩展模块许可或 ArcGIS Engine 运行时空间分析选项许可。

# 实验四 数据组织

实验二中在没有写代码的情况下，生成了一个地图浏览小程序。这个实验我们学习使用代码的方式进行数据组织。

## 一、实验目的

掌握 ArcGIS Engine+VB 数据组织功能的开发

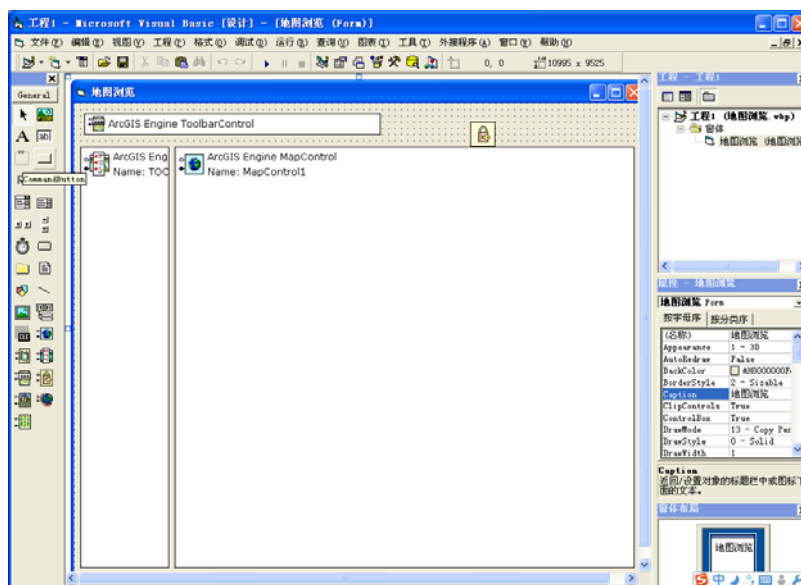
## 二、实验任务

通过 ArcGIS Engine+VB 简单的代码编写，实现添加数据、移除数据等功能的开发。

## 三、实验内容

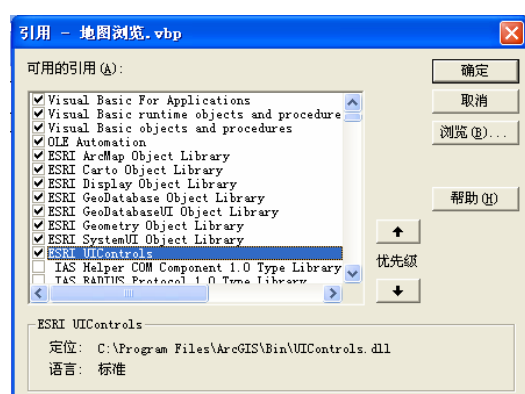
（一）打开实验一中保存的工程“地图浏览”；





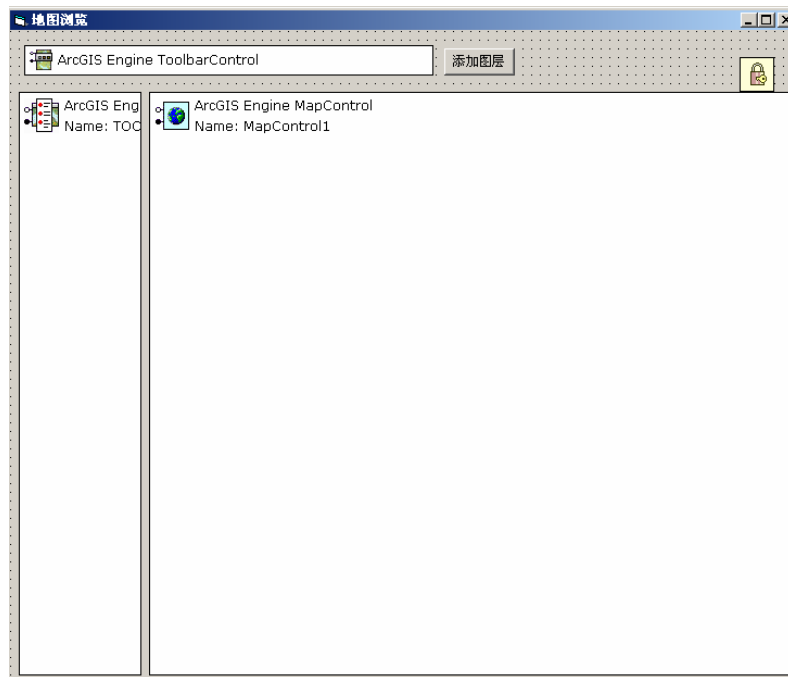
(二) 添加应用：为使 VB 环境中具有 AE 的类库，必须添加 ESRI 的相关应用，比如：ESRI SystemUI Object Library、ESRI System Object Library、ESRI Carto Object Library、ESRI Display Object Library、ESRI Geodatabase Object Library、ESRI Control Commands Object Library、ESRI Output Object Library、ESRI Geometry Object Library 等。

从菜单“工程”—引用添加一下引用：

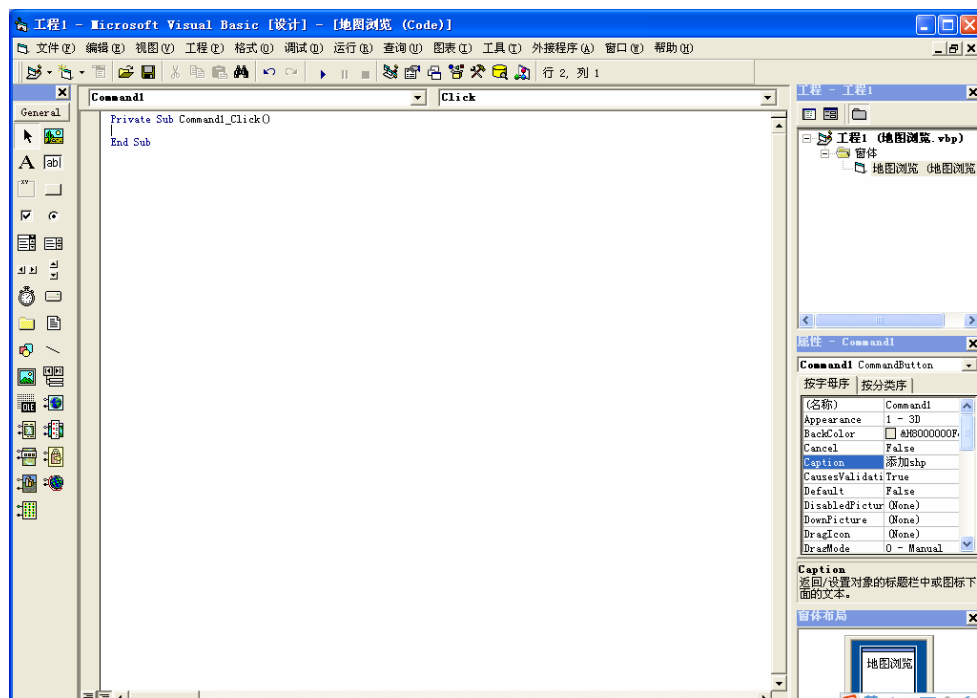


### (三) 添加图层功能的开发

1、向当前窗体中添加按钮，并将该按钮的 Caption 属性设置为“添加图层”，如下图所示



2、双击该按钮，进入代码编写窗口，如下图所示



3、对 Command1 的 Click 事件编写代码，实现添加图层的功能。

```

Private Sub Command1_Click()
    Dim pworkspacefactory As IWorkspaceFactory '定义工作空间
    Dim pfeatworkspace As IFeatureWorkspace '定义对象空间
    Dim pfeatcls As IFeatureClass '定义图斑类，控制图斑行为和属性
    Dim pfeatlayer As IFeatureLayer '定义图斑图层
    '-----
    Set pMap = Me.MapControl1.ActiveView.FocusMap '获得当前map
    'ShapeFile的路径和shp图层名称，你可以用GetDialog或CommonDialog动态获
    '得，在这里作为演示，直接写出ShapeFile的文件路径，和要加载的图层名称。

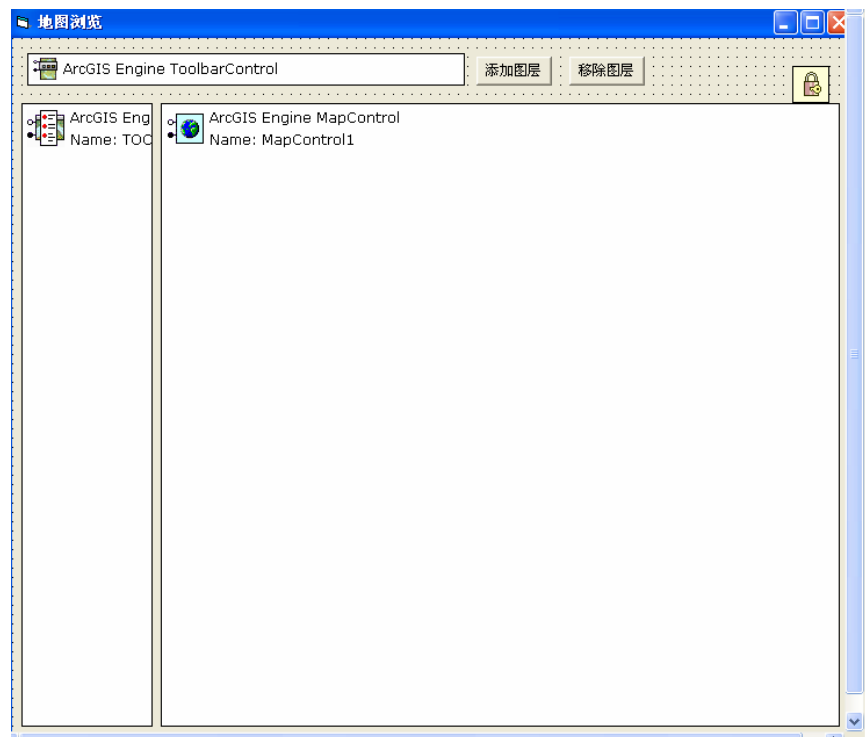
    Dim workspacepath As String: workspacepath = "E:\AE\" '定义并获得工作路径

    '-----
    Set pworkspacefactory = New esriDataSourcesFile.ShapefileWorkspaceFactory '创建shapefile工作空间
    If pworkspacefactory.IsWorkspace(workspacepath) Then
        Set pfeatworkspace = pworkspacefactory.OpenFromFile(workspacepath, 0) '从工作空间获得对象空间
        Set pfeatcls = pfeatworkspace.OpenFeatureClass("COUNTIES") '从对象空间获得图斑类
        Set pfeatlayer = New esriCarto.FeatureLayer '创建新的图斑图层
        pfeatlayer.Name = "COUNTIES"
        Set pfeatlayer.FeatureClass = pfeatcls
        pMap.AddLayer pfeatlayer
    End If
    '直接添加shp文件,这是更简便的做法
    Me.MapControl1.ActiveView.FocusMap.AddLayer "E:\AE\", "COUNTIES"
End Sub

```

#### (四) 移除图层功能的开发

1、向当前窗体中添加按钮，并将该按钮的 Caption 属性设置为“移除图层”，如下图所示



2、双击该按钮，进入代码编写窗口，开发移除图层功能，如下图所示。

```

Private Sub Command2_Click()
End Sub

```

3、对代码进行整理，最终代码窗口为：

```

Dim pMap As IMap
Dim player As ILayer
Dim pIndex As Integer
Dim pOther As IUnknown
Dim pItem As esriTOCControlItem

Private Sub Command1_Click()
Dim pWorkspaceFactory As IWorkspaceFactory '定义工作空间
Dim pFeatWorkspace As IFeatureWorkspace '定义对象空间
Dim pFeatCls As IFeatureClass '定义图斑类，控制图斑行为和属性
Dim pFeatLayer As IFeatureLayer '定义图斑图层
'=====
Set pMap = Me.MapControl1.ActiveView.FocusMap '获得当前map
'ShapeFile的路径和shp图层名称，你可以用GetDialog或CommonDialog动态获
'得，在这里作为演示，直接与ShapeFile的文件路径性，和要加载的图层名称。


Dim sWorkspacePath As String: sWorkspacePath = "E:\AE\ " '定义并获得工作路径

'=====
Set pWorkspaceFactory = New esriDataSourcesFile.ShapefileWorkspaceFactory '创建shapefile工作空间
If pWorkspaceFactory.IsWorkspace(sWorkspacePath) Then
Set pFeatWorkspace = pWorkspaceFactory.OpenFromFile(sWorkspacePath, 0) '从工作空间获得对象空间
Set pFeatCls = pFeatWorkspace.OpenFeatureClass("COUNTIES") '从对象空间获得图斑类
Set pFeatLayer = New esriCarto.FeatureLayer '创建新的图斑图层
pFeatLayer.Name = "COUNTIES"
Set pFeatLayer.FeatureClass = pFeatCls
pMap.AddLayer pFeatLayer
End If
'直接添加shp文件,这是更便捷的做法
Me.MapControl1.ActiveView.FocusMap.AddLayer "E:\AE\ ", "COUNTIES"
End Sub

Private Sub Command2_Click()
If player Is Nothing And pItem = esriTOCControlItem.Layer Then
MsgBox "请选择要删除的图层"
Else
pMap.DeleteLayer player
End If
End Sub
End Sub
Private Sub TOCControl_OnMouseDown(ByVal button As Long, ByVal shift As Long, ByVal X As Long, ByVal Y As Long)
If button = vbLeftButton Then
Me.TOCControl.HitTest X, Y, pItem, pMap, player, pOther, pIndex '获得在TOC控件上鼠标落下的位置、地图、图层等信息
End If
End Sub

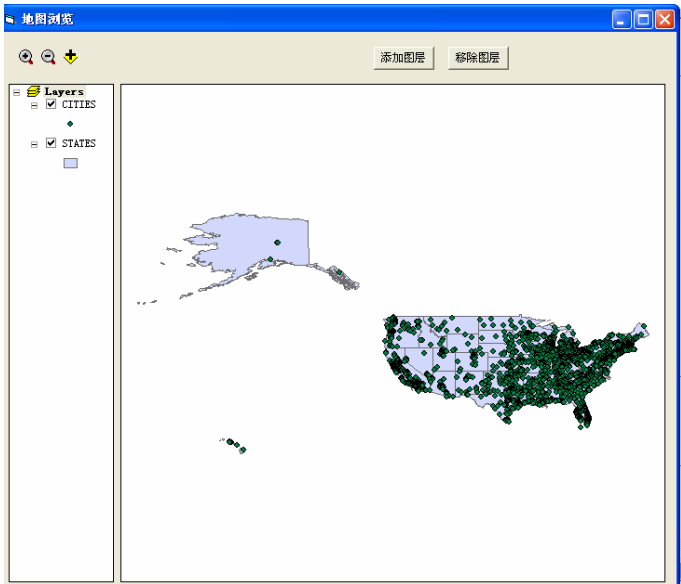
```

### （五）编译运行

点击菜单“运行”——“启动”或工具条中的,



编译运行程序，如下图所示



点击开发按钮，调试运行。

### 四、练习题

练习使用磁盘、文件夹和文件列表框动态获得组织数据所在的路径和文件名。

# 实验五 使用 Itool 和类模块

## 一、实验目的

掌握 Itool 工具的应用和类模块的创建和使用

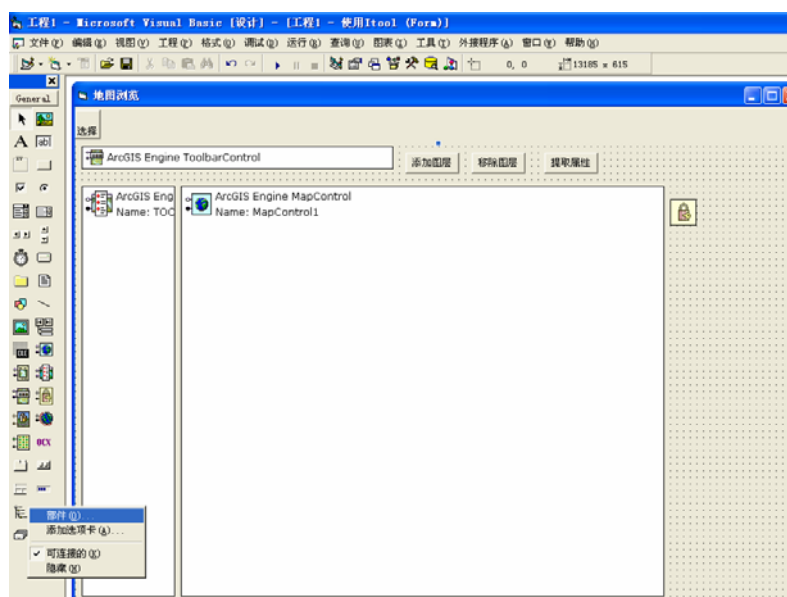
## 二、实验任务

通过 ArcGIS Engine+VB 简单的代码编写,实现 Itool 工具的创建、“选择图斑”功能的开发及相应类模块的创建和使用。

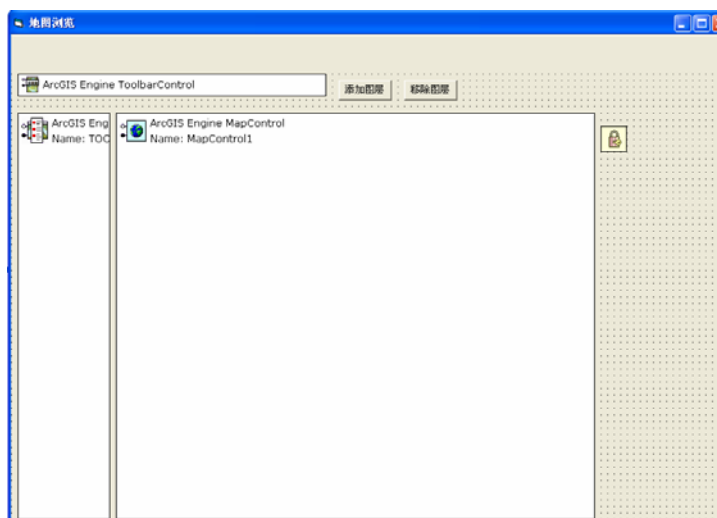
## 三、实验内容

### (一) 创建 Itool 工具

- 1、打开上节实验所做的工程;
- 2、添加 “Microsoft Windows Common Controls 5.0” 部件;

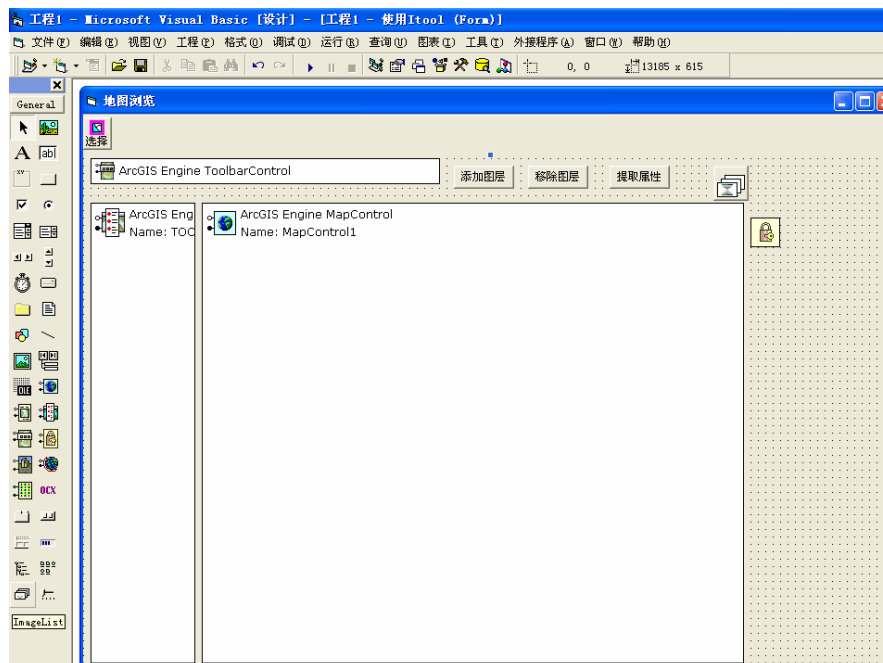


3、选择 “Toolbar” 工具，并在窗体适当位置拖出，窗体布局如图所示：



4、在 toolbar1 中，放置按钮。

(1) 选择 imagelist 控件，并拖放在窗体合适位置



(2) 打开 imagelist1 控件的 “自定义 “属性，



(3) 选择“插入图片”，找到你想用的图片，比如到 ArcGIS 安装路径下找。

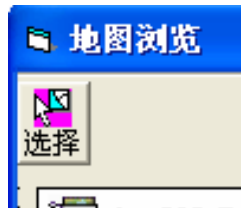
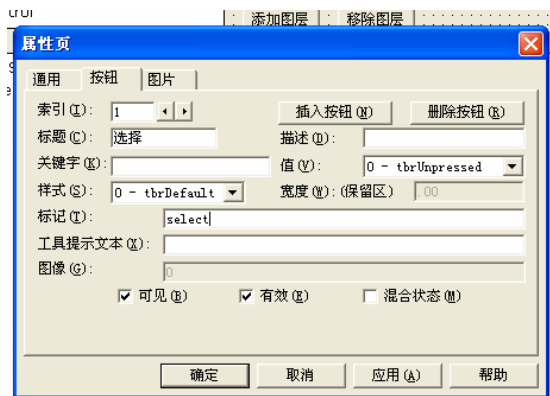


(4) 选中控件 toolbar1，在 toolbar1 的属性窗口，点击自定义



弹出如下对话框，

选择“按钮”选项卡，单击“插入按钮”，输入标题“选择”，标记“select”，图像输入“1”或者“2”这样，一个选择工具就创建好了。

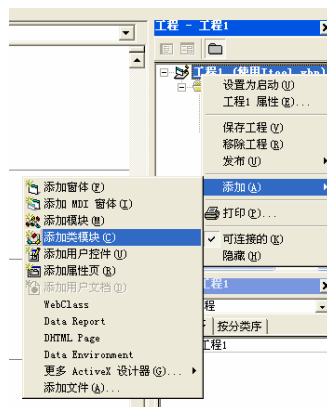


## (二) 类模块的创建及“选择图斑”功能的开发

类模块与窗体模块类似，只是没有可见的用户界面，可以使用类模块创建含有属性和方法的自己的对象。本实验通过创建类模块，创建选择图斑对象，实现选择图斑功能。

### 1、创建新的类模块，并命名为“Classelect”

在工程窗口，选中当前工程点右键，选择添加一类模块

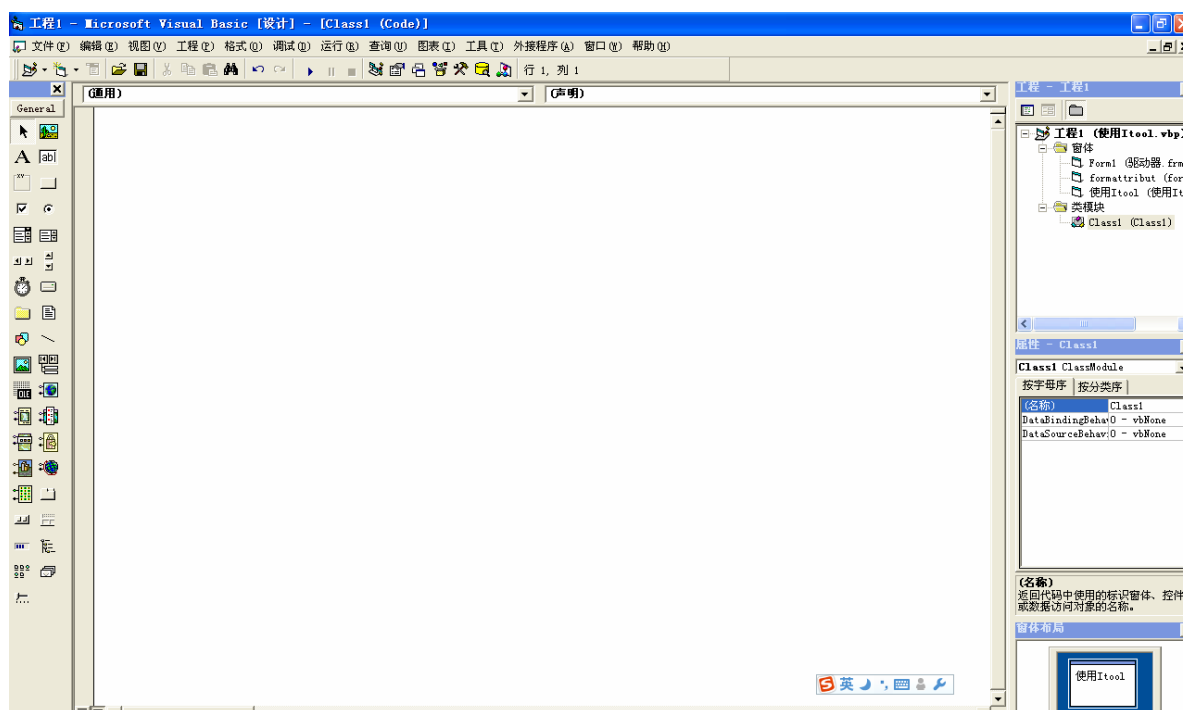


弹出如下对话框，选择“新建”——“类模块”，然后点击打开

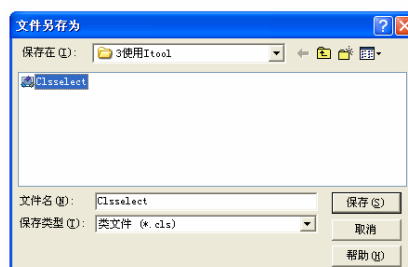




## 创建类模块后的代码窗口



在该类模块的属性窗口，修改其名称属性为“Clsselect”；在工程窗口，选中新创建的类模块，右键选择“保存 Class1”，弹出如下对话框，将该类模块命名为“Clsselect”，保存在实验路径下。



2、“选择图斑”功能的开发：在新创建的类模块代码窗口拷入如下代

码，实现选择图斑的功能。

```
Option Explicit

Implements ICommand
Implements ITool

Private m_pmap As IMap
Private m_pFeedbackEnv As INewEnvelopeFeedback
Private m_pPoint As IPoint
Private m_bIsMouseDown As Boolean

Private Property Get ICommand_Bitmap() As esriSystem.OLE_HANDLE
    ICommand_Bitmap = frmResources.imlBitmaps.ListImages(4).Picture
End Property

Private Property Get ICommand_Caption() As String
    ICommand_Caption = "Select"
End Property

Private Property Get ICommand_Category() As String
    ICommand_Category = "Developer Samples"
End Property

Private Property Get ICommand_Checked() As Boolean
    ICommand_Checked = False
End Property

Private Property Get ICommand_Enabled() As Boolean
    ICommand_Enabled = True
End Property

Private Property Get ICommand_HelpContextID() As Long
    ' No help implemented for this tool
End Property

Private Property Get ICommand_HelpFile() As String
    ' No help implemented for this tool
End Property

Private Property Get ICommand_Message() As String
    ICommand_Message = "Selects features in the map by rectangle"
End Property
```

```

Private Property Get ICommand_Name() As String
    ICommand_Name = "Developer Samples_Select"
End Property

Private Sub ICommand_OnClick()

End Sub

Private Sub ICommand_OnCreate(ByVal hook As Object)
    Set m_pmap = hook.Map
End Sub

Private Property Get ICommand_Tooltip() As String
    ICommand_Tooltip = "Select Features"
End Property

Private Property Get ITool_Cursor() As esriSystem.OLE_HANDLE
    If Not m_bIsMouseDown Then ' not in the middle of rubber banding
        ' ITool_Cursor = frmResources.imlIcons.ListImages(7).Picture
    Else
        ' ITool_Cursor = frmResources.imlIcons.ListImages(8).Picture
    End If
End Property

Private Function ITool_Deactivate() As Boolean
    ITool_Deactivate = True
End Function

Private Function ITool_OnContextMenu(ByVal X As Long, ByVal Y As Long) As
Boolean
End Function

Private Sub ITool_OnDbClick()

End Sub

Private Sub ITool_OnKeyDown(ByVal keyCode As Long, ByVal Shift As Long)

End Sub

Private Sub ITool_OnKeyUp(ByVal keyCode As Long, ByVal Shift As Long)

End Sub

```

```

Private Sub ITool_OnMouseDown(ByVal Button As Long, ByVal Shift As Long,
ByVal X As Long, ByVal Y As Long)
    ' Get the ActiveView for the map

    Dim pactiveview As IActiveView
    Set pactiveview = m_pmap

    'Store current point, set mousedown flag
    Set m_pPoint =
pactiveview.ScreenDisplay.DisplayTransformation.ToMapPoint(X, Y)
    m_bIsMouseDown = True
End Sub

```

```

Private Sub ITool_OnMouseMove(ByVal Button As Long, ByVal Shift As Long,
ByVal X As Long, ByVal Y As Long)
    On Error GoTo ErrorHandler
    If Not m_bIsMouseDown Then Exit Sub
    ' Get the ActiveView for the map
    Dim pactiveview As IActiveView
    Set pactiveview = m_pmap
    ' Create a rubber banding box, if it hasn't been created already
    If (m_pFeedbackEnv Is Nothing) Then
        Set m_pFeedbackEnv = New NewEnvelopeFeedback
        Set m_pFeedbackEnv.Display = pactiveview.ScreenDisplay
        m_pFeedbackEnv.Start m_pPoint
    End If
    'Store current point, and use to move rubberband
    Set m_pPoint =
pactiveview.ScreenDisplay.DisplayTransformation.ToMapPoint(X, Y)
    m_pFeedbackEnv.MoveTo m_pPoint

    Exit Sub
ErrorHandler:
    MsgBox "An error has occured within the Zoom Out Tool." & vbCrLf & vbCrLf
    & _
        "Error Details : " & Err.Description, vbExclamation + vbOKOnly,
    "Error"
End Sub

```

```

Private Sub ITool_OnMouseUp(ByVal Button As Long, ByVal Shift As Long,
ByVal X As Long, ByVal Y As Long)
    Dim pEnv As IEnvelope
    Dim pactiveview As IActiveView

```

```

On Error GoTo ErrorHandler

' Get the ActiveView for the map

Set pactiveview = m_pmap

' Refresh the selections
pactiveview.PartialRefresh esriViewGeoSelection, Nothing, Nothing

' If user dragged an envelope...
If (Not m_pFeedbackEnv Is Nothing) Then
    ' Use it to calculate new extent
    Set pEnv = m_pFeedbackEnv.Stop
    ' Select all feature that intersect with that shape
    m_pmap.SelectByShape pEnv, Nothing, False
' Else...
Else
    ' Select by point
    m_pmap.SelectByShape m_pPoint, Nothing, False
End If

' Refresh the selections
pactiveview.PartialRefresh esriViewGeoSelection, Nothing, Nothing

' Reset rubberband and mousedown state
Set m_pFeedbackEnv = Nothing
m_bIsMouseDown = False

Exit Sub

ErrorHandler:
    MsgBox "An error has occurred within the Select Tool." & vbCrLf & vbCrLf &
-
        "Error Details : " & Err.Description, vbExclamation + vbOKOnly,
"Error"
End Sub

Private Sub ITool_Refresh(ByVal hDC As esriSystem.OLE_HANDLE)

End Sub

```

### 3、保存该类模块

(三) 调用 “Clsselect” 类模块，实现 “选择图斑” 操作。

1、双击控件 Toolbar1, 进入代码编辑窗口

2、在代码顶端除原来定义的对象外, 定义如下几类对象

```
Dim cmd As ICommand
```

3、对 toolbar1 的 buttonclick 事件写入如下代码, 实现当标识为 “select “按钮按下时, 调用类模块” Clsselect “.

```
Private Sub Toolbar1_ButtonClick(ByVal Button As ComctlLib.Button)
    If CStr(Button.Tag) = "select" Then
        Set cmd = New Clsselect '新建clsZoomIn的实例
        cmd.OnCreate Me.MapControl1 '初始化该对象
        Set Me.MapControl1.CurrentTool = cmd '把该对象附给MapControl1
    End If
End Sub
```

(四) 保存上述窗体、类模块和工程

#### 四、练习题

在本实验内容的基础上, 增加查询图斑的 Tool1, 调用 AE 自带的查询命令, 实现查询的功能。

## 实验六 使用鹰眼

### 一、实验目的

掌握过程的创建和调用, 实现同一窗体中两个地图控件的数据一致, 达到鹰眼的效果。

### 二、实验任务

通过创建过程使用鹰眼, 并标识视图在地图中的位置, 掌握同一窗体中两个地图控件的运用。

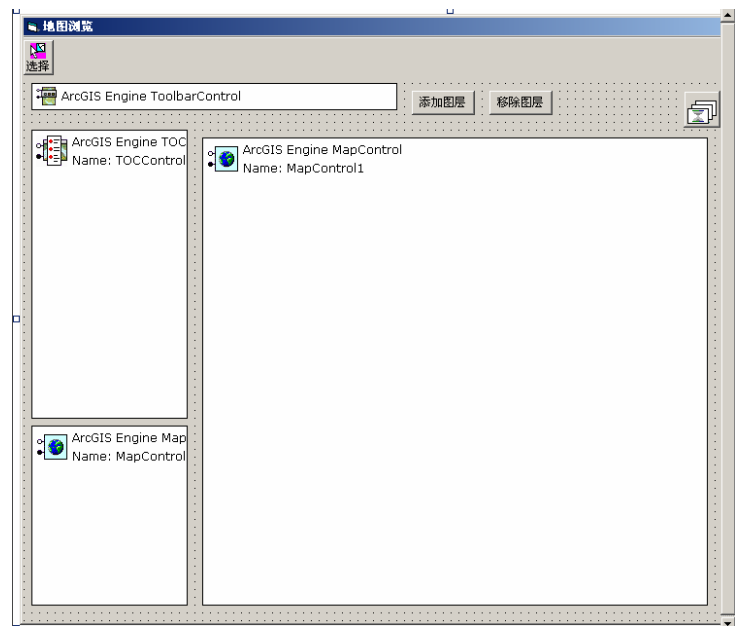
### 三、实验内容

#### (一) 定制窗体

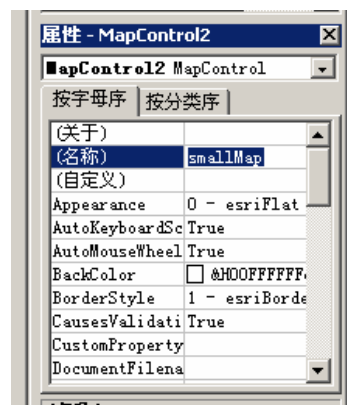
1、打开上节实验所做的工程;

2、在主窗体中添加第 2 个 MapControl 控件, 并调整主窗体布局, 窗

体布局如下图;



3、在 MapControl2 的属性对话框中修改该控件的名称为 smallMap;



(二) 创建过程实现同一窗体中两个地图控件的数据一致

1、在主窗体的代码窗口创建过程 SmallMap-UpdateData，实现 SmallMap 控件和 MapControl1 控件的数据一致，包括添加、删除、移动图层等数据的更新，代码如下：

```
Private Sub SmallMap-UpdateData ()  
  
    Dim i As Integer  
  
    Dim count As Integer  
  
    With Me.MapControl1
```

```

count = .LayerCount

For i = 0 To count - 1

    '这里注意，一定要保证两个地图控件中，图层顺序一致

    Me.smallMap.AddLayer .Layer(i), i

Next

End With

Me.smallMap.Extent = Me.smallMap.FullExtent

End Sub

```

## 2、调用过程 SmallMap\_UpdateData

在主窗体加载和 MapControl1 更新时分别调用过程 SmallMap\_UpdateData，代码如下：

```
Call SmallMap_UpdateData
```

### （三）创建过程用红色方框标识视图在地图中的位置

1、在主窗体的代码窗口创建过程 DrawViewRectInSmallMap，实现在 SmallMap 控件中绘制一个红色的矩形方框，来标识视图在地图中的位置，形成鹰眼，代码如下：

```

Private Sub DrawViewRectInSmallMap(ByVal pEnv As IEnvelope)

    Dim pCurEnv As IEnvelope

    Dim pGContainer As IGraphicsContainer

    Dim pActView As IActiveView

    Dim pElement As IElement

    Dim pFillElement As IFillShapeElement

```



```

Dim pColor As IColor

Dim pOutline As ILineSymbol

Dim pFillSymbol As IFillSymbol


Set pCurEnv = pEnv

Set pGContainer = Me.smallMap.Map

Set pActView = Me.smallMap.ActiveView

pGContainer.DeleteAllElements


Set pElement = New RectangleElement

    pElement.Geometry = pCurEnv


Set pColor = New RgbColor

pColor.RGB = 255 '红色


Set pOutline = New SimpleLineSymbol

With pOutline

    .Width = 1

    .Color = pColor

End With


Set pColor = New RgbColor

```

```
pColor.Transparency = 0
```

```
Set pFillSymbol = New SimpleFillSymbol
```

```
With pFillSymbol
```

```
    .Color = pColor
```

```
    .Outline = pOutline
```

```
End With
```

```
Set pFillElement = pElement
```

```
pFillElement.Symbol = pFillSymbol
```

```
pGContainer.AddElement pElement, 0
```

```
pActView.PartialRefresh esriViewGraphics, Nothing, Nothing
```

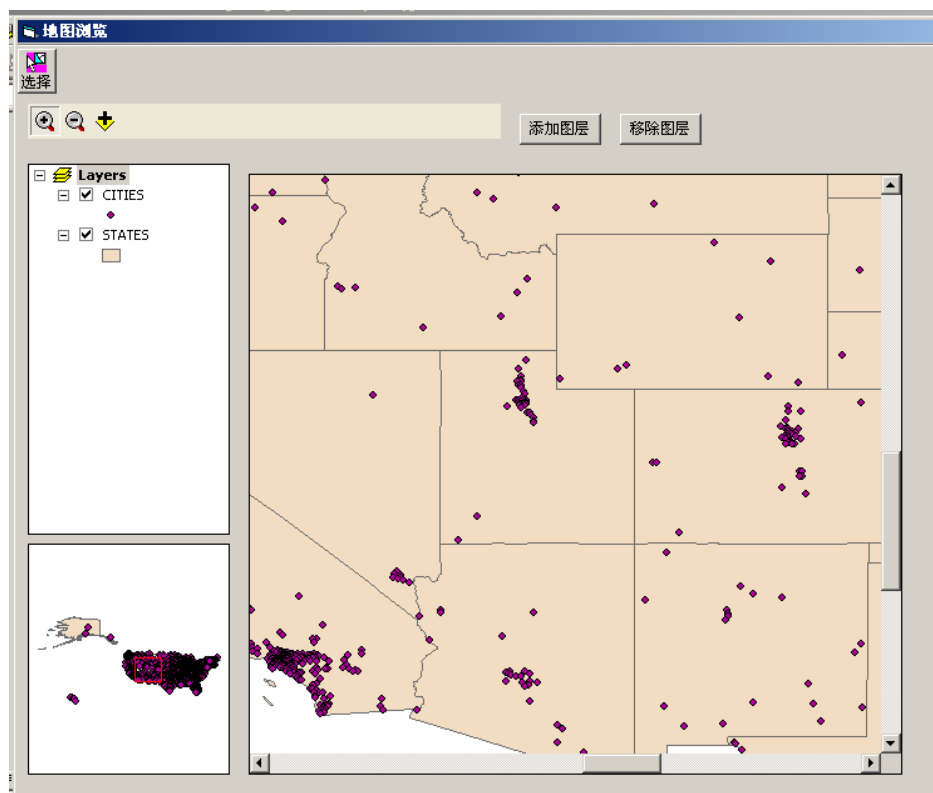
```
End Sub
```

2、传递主图的视图范围参数，调用过程 DrawViewRectInSmallMap

在 MapControl1 更新时调用过程 DrawViewRectInSmallMap，代码如下：

```
Call DrawViewRectInSmallMap(newEnvelope)
```

3、运行程序，如下图所示



#### 四、练习题

- 1、对实验六中创建的 2 个过程加注程序内部文档，包括过程名称、主要功能、参数含义、使用方法、调用该过程的过程/模块等。
- 2、对实验六中的程序段加注程序内部文档，说明各程序段的功能。

## 实验七 应用系统开发

### 一、实验目的

通过应用性 GIS 开发，掌握 ArcGisEngine+VB6 的综合运用能力。

### 二、实验任务

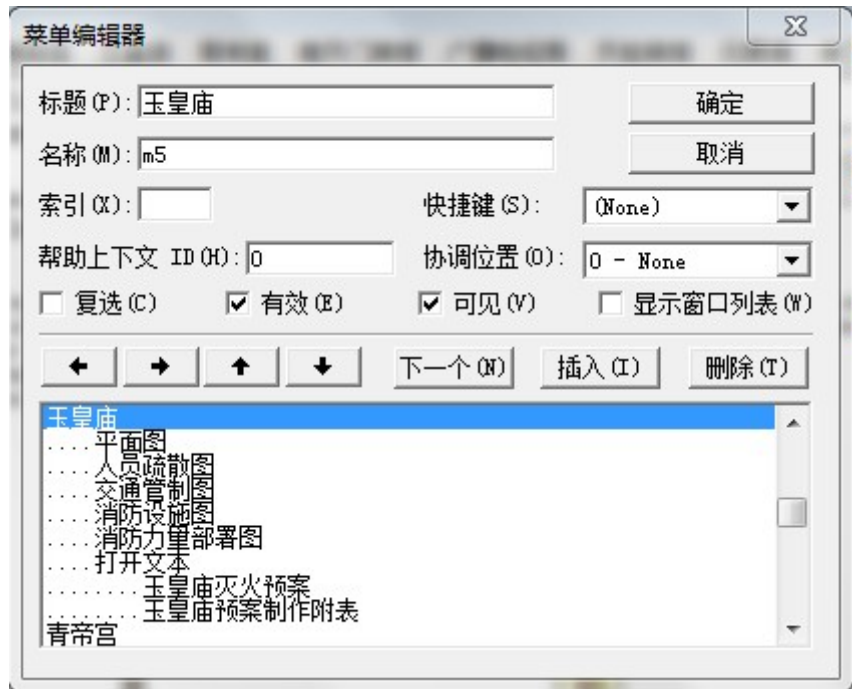
综合运用所学知识，结合泰山消防业务需求，建立一个具有打开地图、文档及 GIS 基本功能的小型消防系统。

### 三、实验内容

#### （一）定制窗体

- 1、定制菜单：通过在窗体上单击鼠标右键弹出的快捷菜单打开菜单

编辑器，在菜单编辑器中编辑菜单。如下图所示：

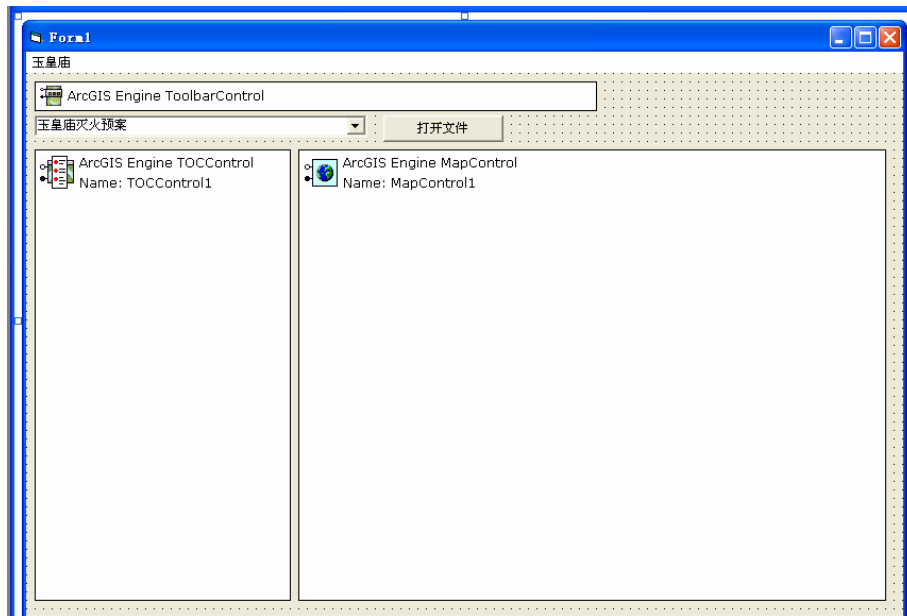


2、加载 AE 控件及相应应用，并在窗体适当位置加载 licenseControl、ToolbarControl、MapControl、TOCControl 控件，并对 ToolbarControl 添加放大、缩小、选择、查询等常用 GIS 功能。其中选择是必选功能。

3、在 ToolbarControl11 控件下添加一个 ComboBOX 控件和一个 CommandButton 控件。将 CommandButton 的 Caption 属性改为“打开文件”，将 ComboBOX 的 TEXT 属性改为玉皇庙灭火预案。如图：



整个窗体布局图如下：



## (二) 打开地图及文档功能的开发

1、定义通用变量。开始编写代码，首先在通用声明中定义变量如下：

```
Dim Hay As Object
```

```
Dim sFileName As String
```

2、加载初始地图。在 Form1 的 Load 事件下写入下列代码，用来实现打开消防系统是加载初始地图。

```
sFileName = "E:\AE\5 应用系统\总体方位图\总体方位图.mxd"
```

```
If MapControl1.CheckMxFile(sFileName) Then
```

```
    MapControl1.LoadMxFile sFileName
```

```
End If
```

3、根据菜单打开相应地图。在菜单的每一项中分别放入下列代码，用来实现点击菜单打开相应地图的功能。

```
sFileName = "E:\AE\5 应用系统\5 玉皇庙\平面图\玉皇庙平面图.mxd"
```

```
MapControl1.LoadMxFile sFileName
```

```
MapControl1.Extent = MapControl1.FullExtent
```

```
Combo1.Clear
```

```
Combo1.Text = "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙预案制作附表"
```

```
sFileName = "E:\AE\5 应用系统\5 玉皇庙\人员疏散图\人员疏散  
图.mxd"
```

```
MapControl1.LoadMxFile sFileName
```

```
MapControl1.Extent = MapControl1.FullExtent
```

```
Combo1.Clear
```

```
Combo1.Text = "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙预案制作附表"
```

```
sFileName = "E:\AE\5 应用系统\5 玉皇庙\交通管制图\交通管  
制.mxd"
```

```
MapControl1.LoadMxFile sFileName
```

```
MapControl1.Extent = MapControl1.FullExtent
```

```
Combo1.Clear
```

```
Combo1.Text = "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙预案制作附表"
```

```
sFileName = "E:\AE\5 应用系统\5 玉皇庙\消防设施图\消防设  
施.mxd"
```

```
MapControl1.LoadMxFile sFileName
```

```
MapControl1.Extent = MapControl1.FullExtent
```

```
Combo1.Clear
```

```
Combo1.Text = "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙预案制作附表"
```

```
sFileName = "E:\AE\5 应用系统\5 玉皇庙\消防力量部署图\消防力  
量部署.mxd"
```

```
MapControl1.LoadMxFile sFileName
```

```
MapControl1.Extent = MapControl1.FullExtent
```

```
Combo1.Clear
```

```
Combo1.Text = "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙灭火预案"
```

```
Combo1.AddItem "玉皇庙预案制作附表"
```

```
Set Hay = CreateObject("Word.Application")
```

```
Hay.Visible = True
```

```
Hay.Documents.Open FileName:="E:\AE\5 应用系统\玉皇庙灭火预  
案（打印）.DOC", ReadOnly:=False
```

```
Set Hay = CreateObject("Word.Application")
```

```
Hay.Visible = True
```

```
Hay.Documents.Open FileName:="E:\AE\5 应用系统\青预案制作  
附.DOC", ReadOnly:=False
```

此时点击菜单后 ComboBOX 中会加载两个项目，分别为“玉皇庙灭火预案”和“玉皇庙预案制作附表”。

4、打开相应 Word 文档：在 CommandButton 的 Click 事件中写入下列代码，用来打开相应的文档。

```
Set Hay = CreateObject("Word.Application")
```

```
Hay.Visible = True
```

```
If Combo1.Text = "玉皇庙灭火预案" Then
```

```
Hay.Documents.Open FileName:="E:\AE\5 应用系统\玉皇庙灭火  
预案（打印）.DOC", ReadOnly:=False
```

```
End If
```

```
If Combo1.Text = "玉皇庙预案制作附表" Then
```

```
Hay.Documents.Open FileName:="E:\AE\5 应用系统\玉预案制作  
附表.DOC", ReadOnly:=False
```

```
End If
```

（三）右键菜单功能的开发



在 MapContorl 中的 OnMouseDown 事件中写入下列代码，用来实现选择要素后，右键点击弹出相应菜单的功能。

```
If button = vbRightButton Then

Dim index As Long

Dim pfeature As IFeature

Dim pFeaturelayer As IFeatureLayer

Dim pFeatureCursor As IFeatureCursor

Dim pFeatureSelection As IFeatureSelection

Dim pSelectionSet As ISelectionSet

Set pfeature = New Feature

Set pFeaturelayer = MapControl1.Map.Layer(2)

Set pFeatureSelection = pFeaturelayer

Set pSelectionSet = pFeatureSelection.SelectionSet

If pSelectionSet.Count = 0 Then Exit Sub

pSelectionSet.Search Nothing, False, pFeatureCursor

Set pfeature = pFeatureCursor.NextFeature

index = pfeature.Fields.FindField("Id")

Select Case pfeature.Value(index)

Case 23

Form1.PopupMenu m5

End Select

End If
```

至此，功能已经基本可以完全实现。

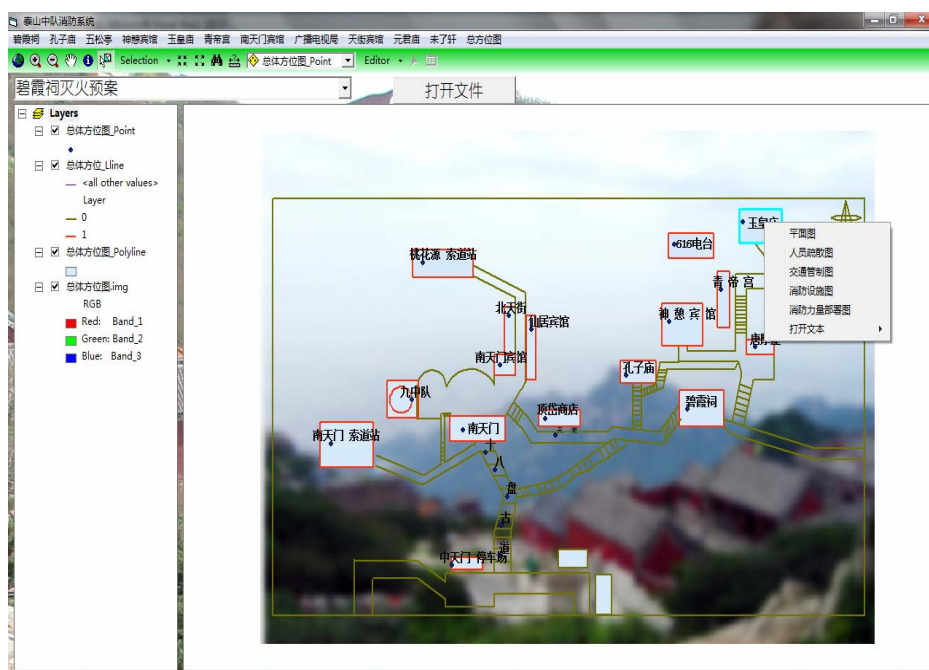
(四) 注意事项:

1、使用右键菜单链接打开地图时，要先用选择工具选中图斑，然后点击右键弹出菜单。

2、在写入代码的时候应该注意代码中各个控件的名字是否与自己工程中控件的名字相同。

3、凡是有文件路径的代码，一定要保证路径与自己所存放的文件路径一致，否则，程序将无法运行。

调试完成后运行程序，运行效果如下图：



#### 四、练习题

根据所学知识，利用已经获得的相关数据，除实验内容中的功能外，开发回到主界面的功能，建立一个小型的泰山消防系统，并上交作业。系统运行界面如下：

