实验九　地图页面布局

9.1 背景知识

9.1.1 PagelayoutControl 控件

页面布局控件 PagelayoutControl 主要用于地图制图，操作各种元素对象和移动版式页面等 ArcMap 布局视图所实现的功能，也可以检查和载入地图文档。PagelayoutControl 控件中封装了页面布局 PageLayout 组件类，它提供了在布局试图中控制元素的属性和方法。PageLayout 的属性主要有：（1）Printer，处理地图打印时的一系列设定；（2）Page，处理控件的页面效果；（3）Element，管理控件中的地图元素，PageLayout 及相关对象结构关系如图 9-1 所示。

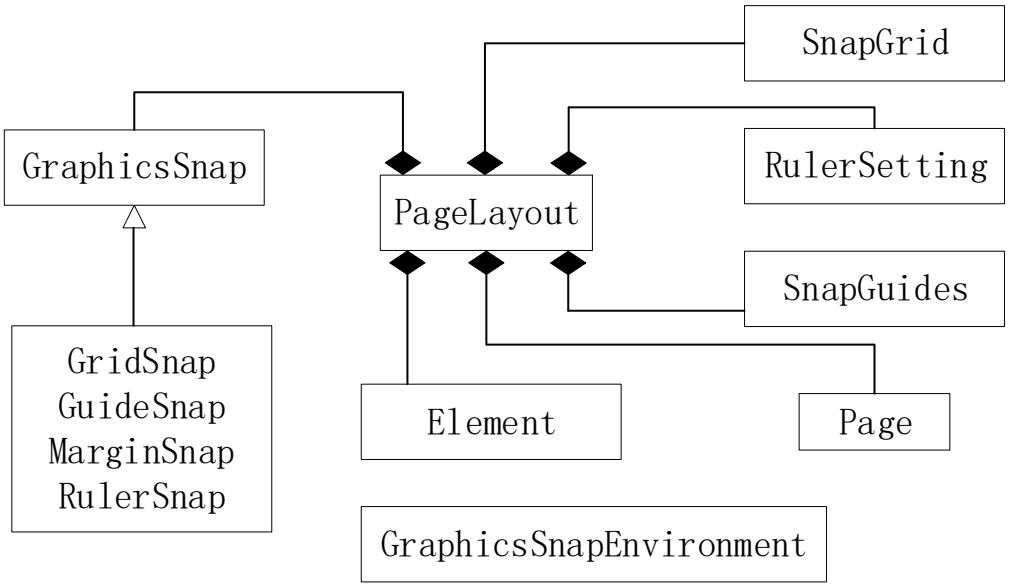

图 9-1　PageLayout 及相关对象结构关系

9.1.2 地图元素对象

地图元素 Element 主要分为两类：（1）图形元素 GraphicElement，作为图形的形式而存在；（2）框架元素 FrameElement，作为不可见的容器而存在，地图元素的主要对象如图 9-2 所示。

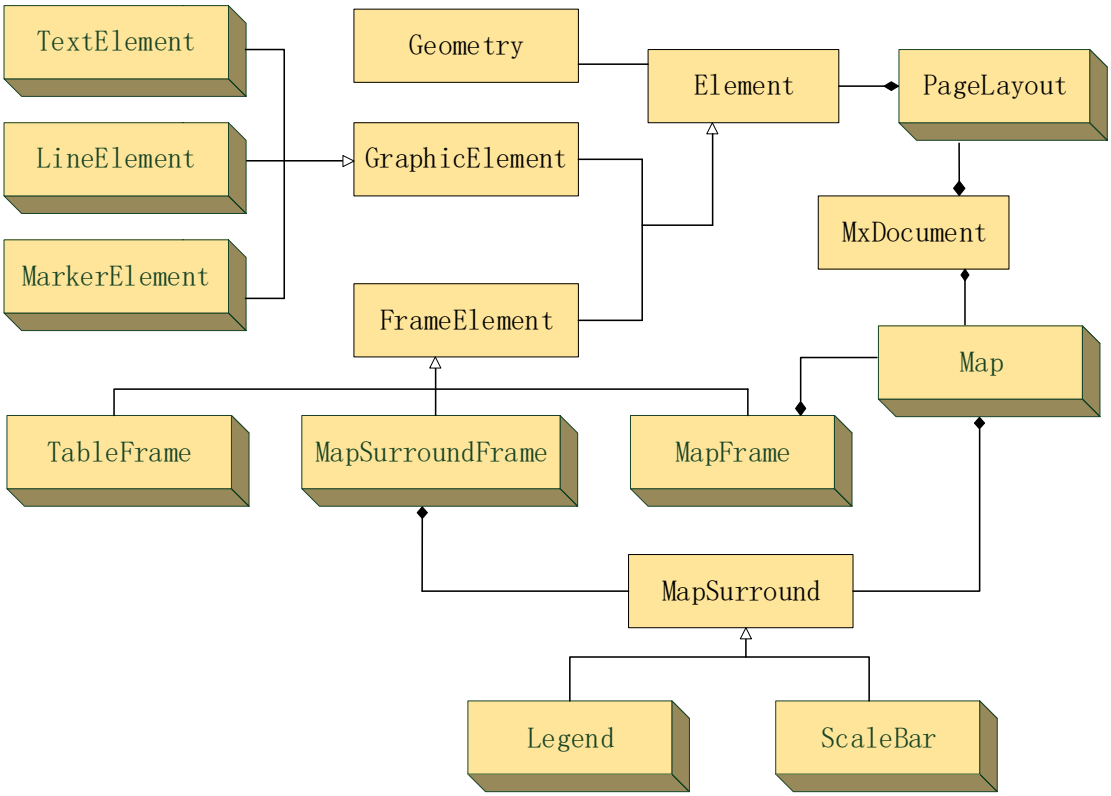

图 9-2　地图元素的主要对象

框架元素 FrameElement 主要包括:(1)地图框架 MapFrame 是 Map 的容器，PageLayout 对象至少拥有一个 MapFrame，每个 MapFrame 拥有一个 Map；可使用样式设置 MapFrame 的边框 Border、阴影 Shadow、背景 Background 和地图网格 MapGrid 属性等。（2）地图整饰框架 MapSurroundFrame 是地图整饰 MapSurround 的容器，MapSurround 是一种与地图对象相关联的特殊地图元素，包括图例 Legend、指北针 NorthArrow、比例尺条 ScaleBar 等。

在 ArcMap 应用程序中的页面布局相关对象如图 9-3 所示。

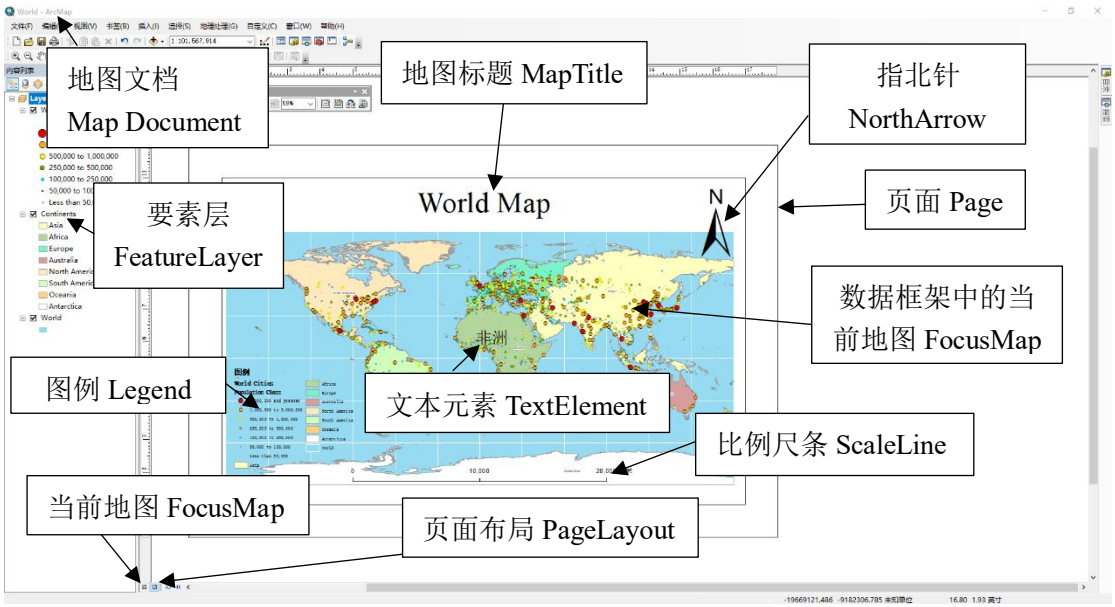

图 9-3　ArcMap 应用程序的页面布局相关对象

9.2 实验目的

（1）掌握 PagelayoutControl/TOCControl/ToolbarControl 应用程序的开发；

（2）熟悉设置地图网格和地图框架的属性（边框、阴影、背景等）；

（3）掌握地图整饰对象（图例、指北针、比例尺）的使用。

9.3 实验内容

（1）使用 PagelayoutControl 控件建立桌面应用程序；

（2）添加和删除地图网格；

（3）设置地图框架的边框、阴影、背景；

（4）添加地图图例、指北针、比例尺等地图整饰对象。

9.4 实验数据

见安装目录：

…\DeveloperKit10.6\Samples\ArcObjectsSampleData\data\California

9.5 实验步骤

9.5.1 建立 PagelayoutControl 应用程序

（1）新建“MapControl Application”项目，为项目命名为“Test9”并设置保存位置。将 axMapControl1 控件的“Dock”属性设置为“None”，向主窗体中添加一个 TabControl 控件和一个 PageLayoutControl 控件。修改“tabControl1”控件的 TabPages 属性，在 TabPage 集合编辑器中将 TabPage1 的“Name”和“Text”属性设置为“TabPageMap”；将 TabPage2 的“Name”和“Text”属性设置为“TabPageLayout”，如图 9-4 所示。

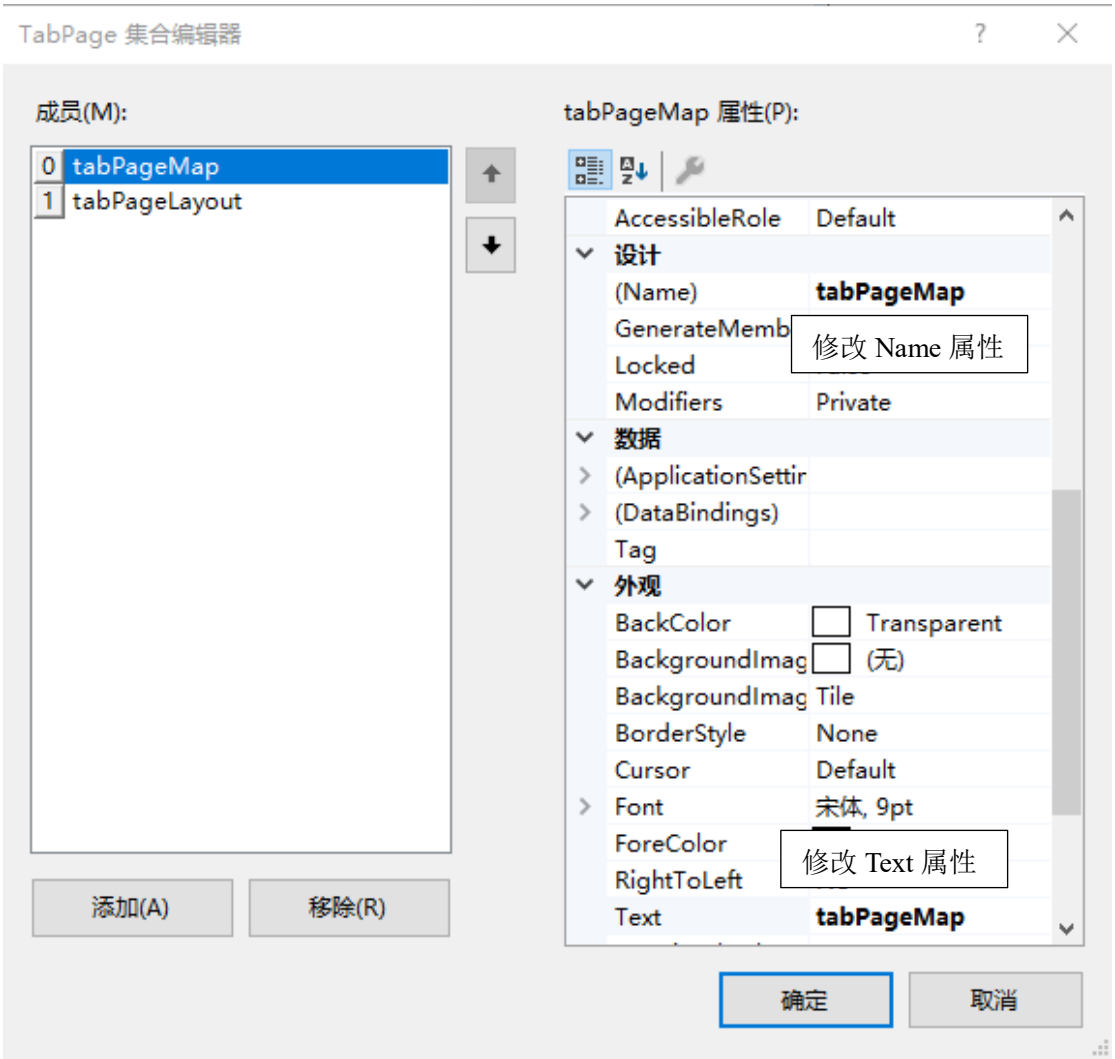

图 9-4　修改 tabControl1 的 TabPages 属性

调整 axMapControl1、axPageLayoutControl1 和 tabControl1 的大小，将 axMapControl1 控件放到 tabControl1 的 tabPage1 中，并设置 axMapControl1 控件的“Dock”属性为“Fill”；将 axPageLayoutControl1 控件放到 tabControl1 的 tabPage2 中，并设置 axMapControl1 控件的“Dock”属性为“Fill”。设置属性完毕后主窗

体界面如图 9-5 所示。

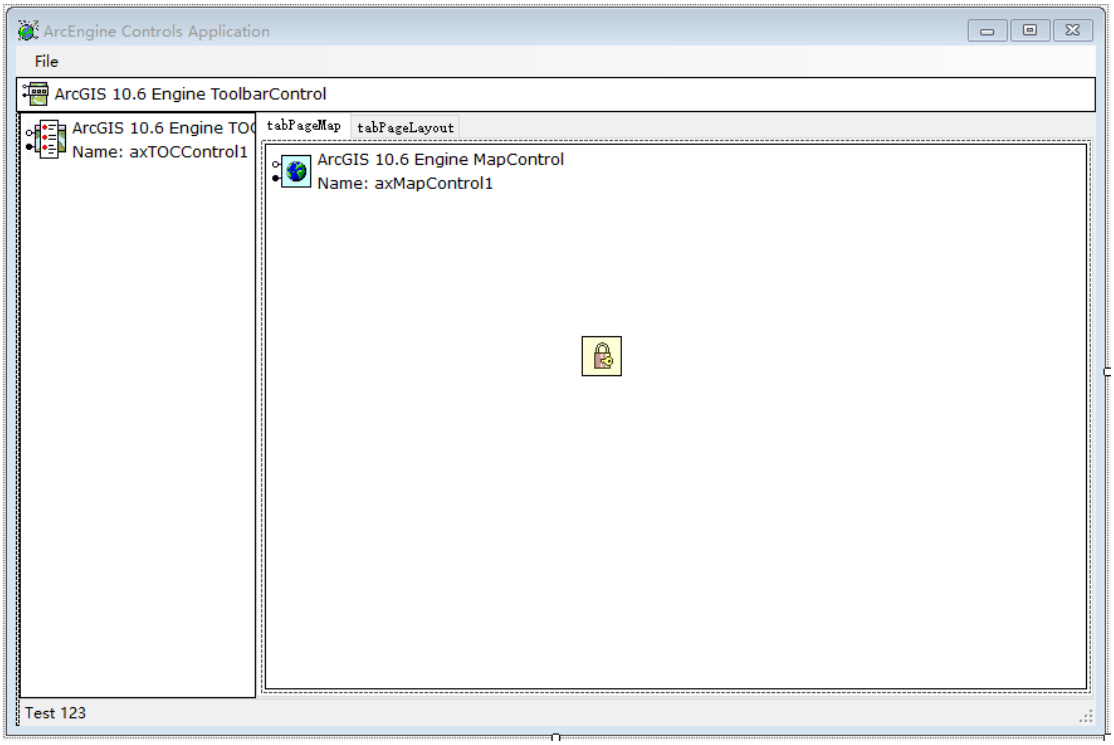

图 9-5　窗体界面设计

在 ToolbarControl 中添加页面布局相关按钮命令，如图 9-6 所示。

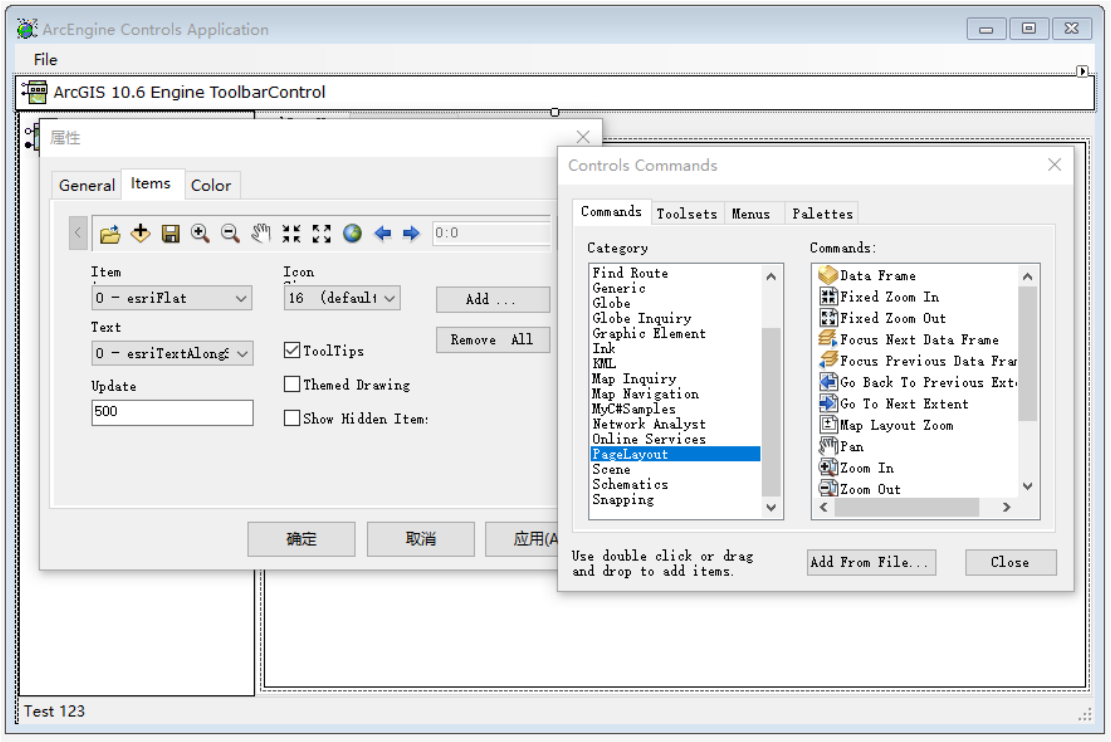

图 9-6　添加 ToolbarControl 页面布局相关按钮

（2）在“MainFrom”主窗体类中添加指向 PagelayoutControl 私有成员变量：

```
private IPageLayoutControl3 m_PageLayoutControl = null;
```

在 MainFrom 窗体加载事件中添加 PageLayoutControl 接口获取的代码：

MainForm.cs（节选）	功能：获取 PageLayoutControl 接口

```
private void MainForm_Load(object sender, EventArgs e)
{
    //get the MapControl
    m_mapControl = (IMapControl3)axMapControl1.Object;
    m_PageLayoutControl =
                        (IPageLayoutControl3)axPageLayoutControl1.Object;
    //disable the Save menu (since there is no document yet)
    menuSaveDoc.Enabled = false;
}
```

（3）在 MainFrom 窗体类代码中为“axPageLayoutControl1”控件拷贝地图的私有成员函数，代码如下：

MainForm.cs（节选）	功能：为 axPageLayoutControl1 控件拷贝地图的私有成员函数

```
private void CopyAndOverwriterMap()
{
    //新建对象拷贝类
    IObjectCopy objectCopy = new ObjectCopyClass();
    //获得主地图对象
    object fromMap = axMapControl1.Map;
    //获得页面布局地图对象
    object objMap = axPageLayoutControl1.ActiveView.FocusMap;
    //将主地图拷贝给页面布局地图
    objectCopy.Overwrite(fromMap, ref objMap);
    //设置页面布局地图范围为地图全局范围
    axPageLayoutControl1.Extent = axPageLayoutControl1.FullExtent;
    //刷新页面布局窗口
    axPageLayoutControl1.Refresh();
}
```

修改 axMapControl1 控件的地图更换事件响应函数，代码如下：

MainForm.cs（节选）	功能：修改 axMapControl1 控件的地图更换事件响应函数

```
private void axMapControl1_OnMapReplaced(object sender,
                                    IMapControlEvents2_OnMapReplacedEvent e)
{
    //get the current document name from the MapControl
    m_mapDocumentName = m_mapControl.DocumentFilename;
    //if there is no MapDocument, diable the Save menu and clear the statusbar
    if (m_mapDocumentName == string.Empty)
    {
```

```
            menuSaveDoc.Enabled = false;
            statusBarXY.Text = string.Empty;
        }
        else
        {
            //enable the Save manu and write the doc name to the statusbar
            menuSaveDoc.Enabled = true;
            statusBarXY.Text = System.IO.Path.GetFileName
                                                    (m_mapDocumentName);
        }
        //拷贝主地图到页面布局窗口
        CopyAndOverwriterMap();
}
```

添加 axMapControl1_OnFullExtentUpdated 函数，在地图全局范围发生改变时，重新拷贝主地图到页面布局，代码如下：

MainForm.cs（节选）	功能：重新拷贝主地图到页面布局

```
private void axMapControl1_OnFullExtentUpdated(object sender,
                                IMapControlEvents2_OnFullExtentUpdatedEvent e)
{
    //拷贝主地图到页面布局窗口
    CopyAndOverwriteMap();
}
```

（4）添加 axPagelayoutControl1 鼠标移动事件响应函数，代码如下：

MainForm.cs（节选）	功能：axPagelayoutControl1 鼠标移动事件响应函数

```
private void axPageLayoutControl1_OnMouseMove(object sender,
                                IPageLayoutControlEvents_OnMouseMoveEvent e)
{
    statusBarXY.Text = string.Format("{0}, {1}   {2}",
            e.pageX.ToString("#######.##"), e.pageY.ToString("#######.##"),
            axPageLayoutControl1.ActiveView.FocusMap.MapUnits.
            ToString().Substring(4));
}
```

添加 tabControl1 控件的标签页选择改变事件响应函数，代码如下：

MainForm.cs（节选）	功能：tabControl1 控件标签页选择改变事件响应函数

```
private void tabControl1_SelectedIndexChanged(object sender, EventArgs e)
{
    if (tabControl1.SelectedIndex == 0)
    {
        //设置axToolbarControl1控件关联axMapControl1控件
        axToolbarControl1.SetBuddyControl(axMapControl1.Object);
```

```
        }
        else
        {
            //设置axToolbarControl1控件关联axPageLayoutControl1控件
            axToolbarControl1.SetBuddyControl(axPageLayoutControl1.Object);
        }
}
```

（5）编译并运行程序，运行结果如图 9-7 所示，基于 MapControl、PagelayoutControl、TOCControl、ToolbarControl 控件的应用程序开发实现。

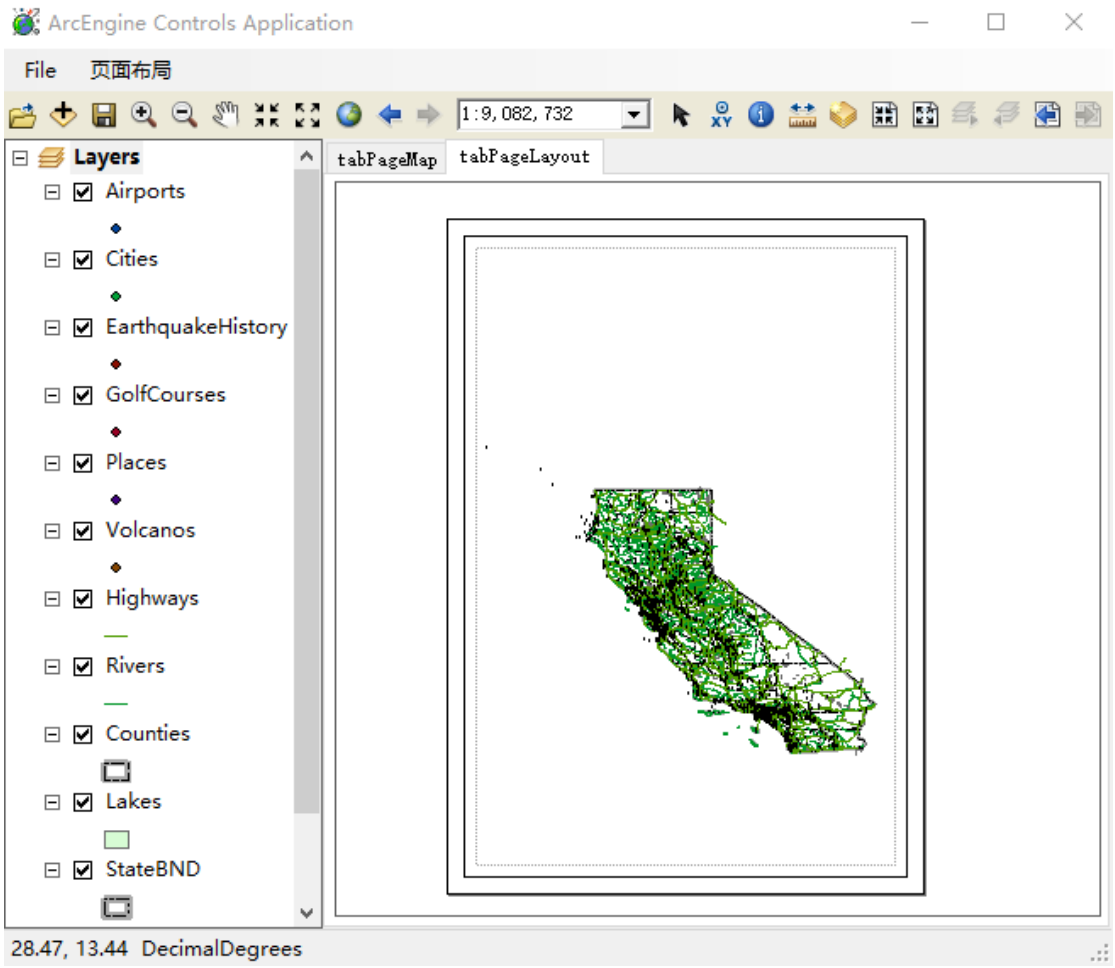

图 9-7　页面布局应用程序运行结果

9.5.2 地图网格操纵

下面主要实现地图页面布局 PageLayout 视图中地图网格 MapGrid 的创建和删除。通过 IMapGrids 和 IMapGrid 接口来实现对 MapGrid 的添加、删除等操作。本质上 IMapGrids 就是 MapFrame 对象，IMapGrids 接口只能被 MapFrame 对象来实现，通过 IMapGrids 接口，可以对一个具体的 MapFrame 所展示的网格进行设置。IMapGrid 是个可以对所有类型网格（Grid）的属性进行设置的接口，四种

类型的 MapGrid 类实现了 IMapGrid 接口，它们是 MeasuredGrid、Graticule、IndexGrid，CustomOverlayGrid。MapGrid 主要由格网线 GridLine、格网标注 GridLabel、和格网边框 GridBorder 三部分组成。

（1）为 Test9 项目添加 “Base Command”按钮命令新建项，按钮命令类命名为“CmdAddMapGrid”。选中项目“Test9”，右键点击【添加】→【新建项】，如图 9-8 所示。

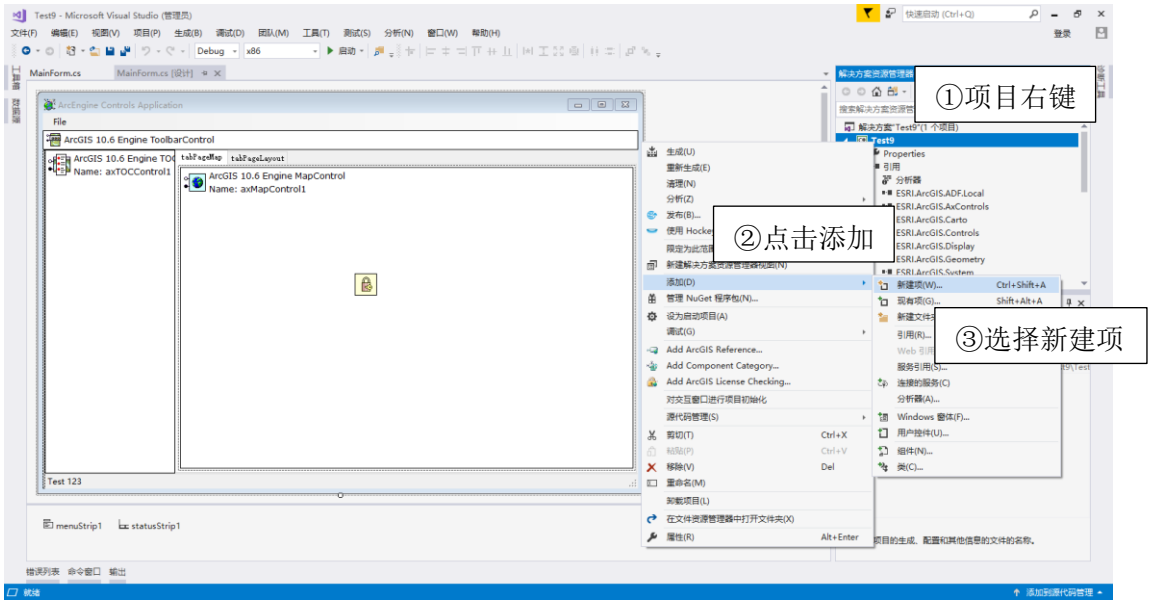

图 9-8　添加新建项

选择【ArcGIS】→【Extending ArcObjects】→【Base Command】，如图 9-9 所示。

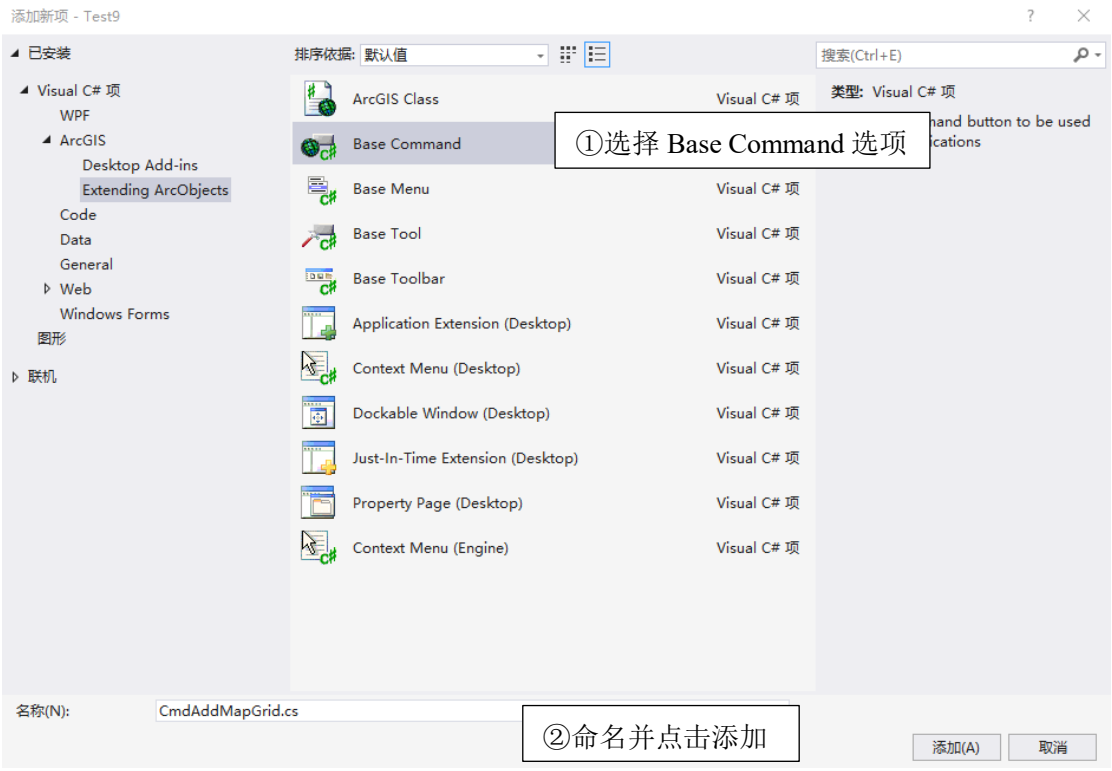

图 9-9　添加“Base Command”按钮命令类

（2）为按钮命令类“CmdAddMapGrid”导入引用：

```
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.esriSystem;
using stdole;
```

为类“CmdAddMapGrid”添加私有成员函数 CreatMeasuredGrid()，并修改 OnClick 事件响应函数代码：

CmdAddMapGrid.cs（节选）	功能：添加生成格网函数并调用

```
public override void OnClick()
{
    // TODO: Add CmdAddMapGrid.OnClick implementation
    CreatMeasuredGrid(m_hookHelper.ActiveView,m_hookHelper.PageLayout);
}
private void CreatMeasuredGrid(IActiveView pActiveView,
                                                IPageLayout pPageLayout)
{
    IMap map = pActiveView.FocusMap;
    IMeasuredGrid pMeasuredGrid = new MeasuredGridClass();
    //设置格网基本属性
    pMeasuredGrid.FixedOrigin = false;
    pMeasuredGrid.Units = map.MapUnits;
```

```
        //纬度间隔
        pMeasuredGrid.XIntervalSize = 5;
        //经度间隔
        pMeasuredGrid.YIntervalSize = 5;
        //设置GridLable格式
        IGridLabel pGridLabel = null;
        IFormattedGridLabel pFormattedGridLabel =
                                                new FormattedGridLabelClass();
        INumericFormat pNumericFormat = new NumericFormatClass();
        pNumericFormat.AlignmentOption =
                                        esriNumericAlignmentEnum.esriAlignLeft;
        pNumericFormat.RoundingOption =
                            esriRoundingOptionEnum.esriRoundNumberOfDecimals;
        pNumericFormat.RoundingValue = 0;
        pNumericFormat.ZeroPad = true;
        pFormattedGridLabel.Format = pNumericFormat as INumberFormat;
        pGridLabel = pFormattedGridLabel as IGridLabel;
        StdFont pStdFont = new stdole.StdFontClass();
        pStdFont.Name = "宋体";
        pStdFont.Size = 25;
        pGridLabel.Font = pStdFont as IFontDisp;
        IMapGrid pMapGrid = pMeasuredGrid as IMapGrid;
        pMapGrid.LabelFormat = pGridLabel;
        //将格网添加到地图上
        IGraphicsContainer graphicsContainer = pPageLayout as IGraphicsContainer;
        IFrameElement frameElement = graphicsContainer.FindFrame(map);
        IMapFrame mapFrame = frameElement as IMapFrame;
        IMapGrids mapGrids = null;
        mapGrids = mapFrame as IMapGrids;
        mapGrids.AddMapGrid(pMapGrid);
        pActiveView.PartialRefresh(esriViewDrawPhase.esriViewBackground,
                                                                null, null);
}
```

（3）在主窗体菜单 menuStrip1 控件上添加一个一级菜单为“页面布局”，在该菜单下添加二级菜单为“创建地图网格”，如图 9-10 所示。

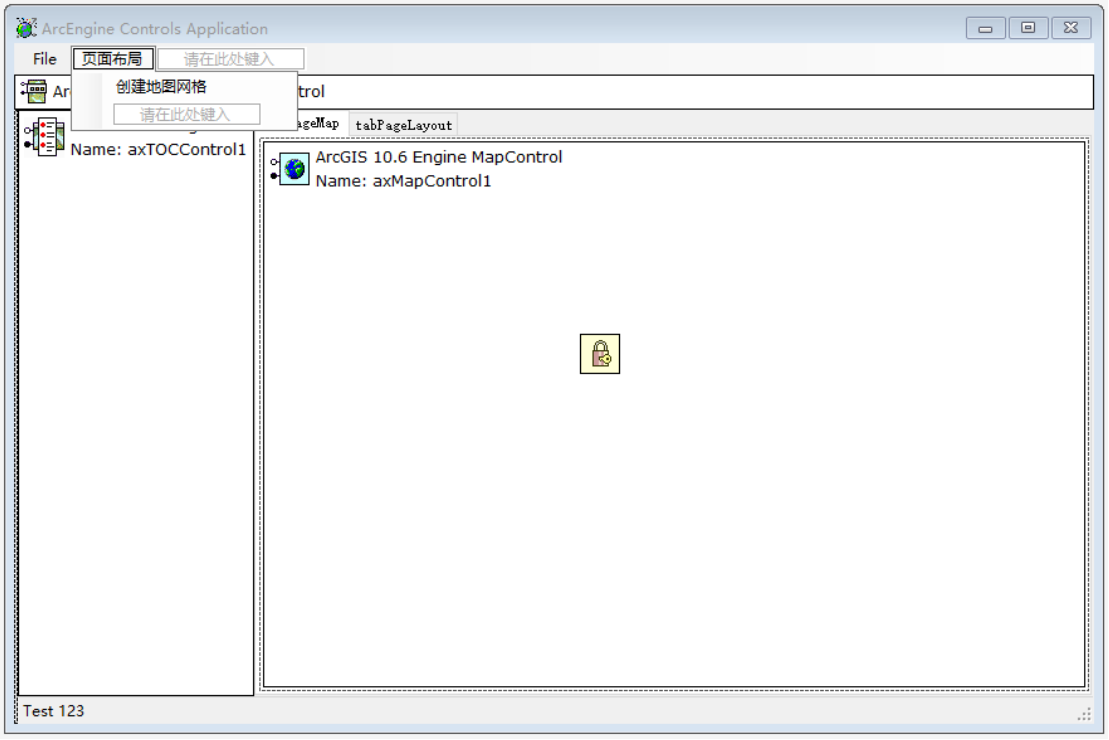

图 9-10　添加“创建地图网格”二级菜单

双击主窗体菜单 menuStrip1 控件二级菜单的“创建地图格网”，为其添加点击事件响应函数，代码如下：

MainForm.cs（节选）	功能：“创建地图网格”菜单项点击事件响应函数

```
private void 创建地图网格ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ICommand command = new CmdAddMapGrid();
    command.OnCreate(m_PageLayoutControl.Object);
    command.OnClick();
}
```

（4）编译并运行程序，添加地图文档，切换视图为 PageLayout 地图，点击“创建地图网格”，运行结果如图 9-11 所示。

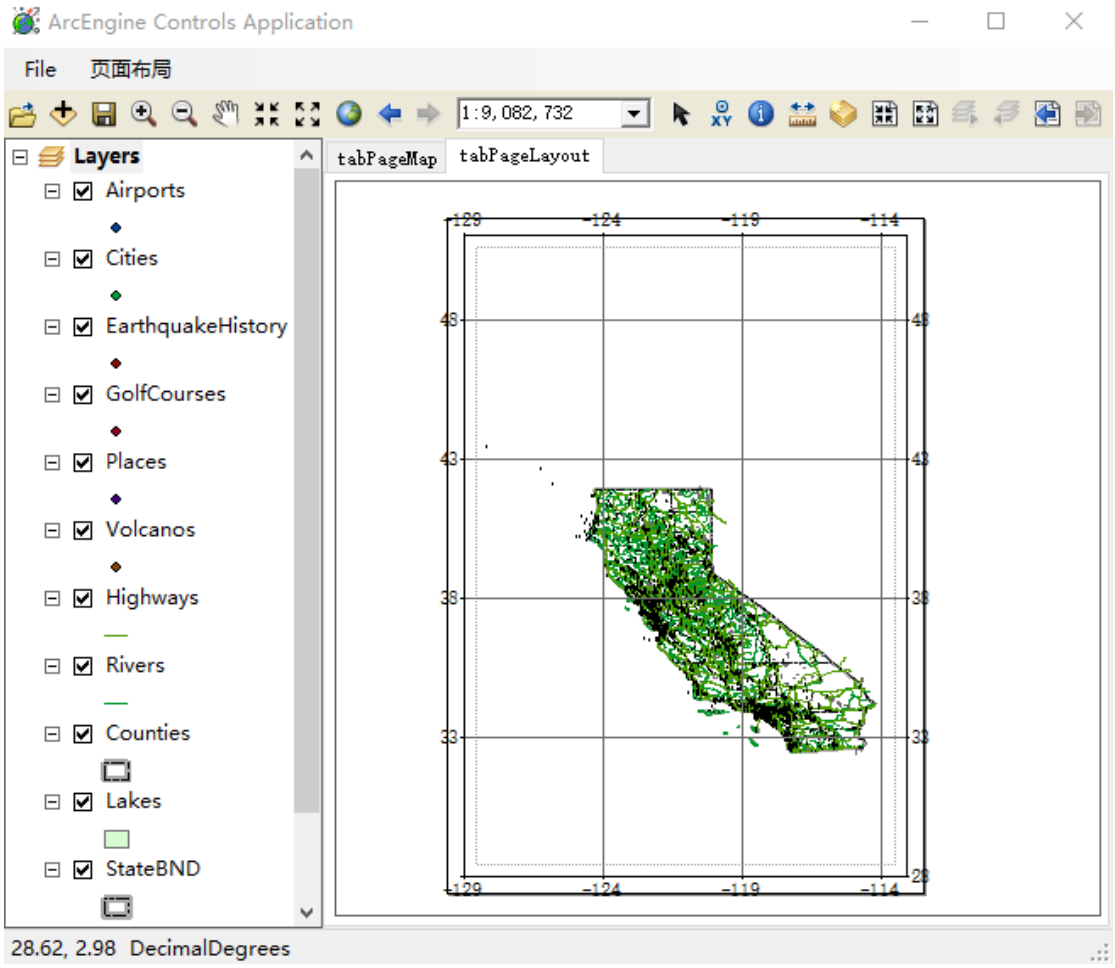

图 9-11　创建地图网格功能运行效果

（5）为 Test9 项目添加新建项“Base Command”按钮命令，按钮命令类命名为“CmdDelMapGrid”。为按钮命令类“CmdDelMapGrid”导入引用：

```
using ESRI.ArcGIS.Carto;
```

为 OnClick 重载事件添加响应函数代码，代码如下：

MainForm.cs（节选）	功能：OnClick 重载事件响应事件

```
public override void OnClick()
{
    // TODO: Add CmdDelMapGrid.OnClick implementation
    IGraphicsContainer gc = m_hookHelper.ActiveView.GraphicsContainer;
    IMap map = m_hookHelper.ActiveView.FocusMap;
    IMapFrame mf = gc.FindFrame(map) as IMapFrame;
    IMapGrids mgs = mf as IMapGrids;
    mgs.ClearMapGrids();
    m_hookHelper.ActiveView.PartialRefresh(
                    esriViewDrawPhase.esriViewGraphics, null, null);
}
```

（6）在主窗体菜单 menuStrip1 控件的一级菜单“页面布局”下添加二级菜单为“删除地图网格”，双击主窗体菜单 menuStrip1 控件二级菜单的“删除地图格网”，为其添加点击事件响应函数，代码如下：

CmdDelMapGrid.cs（节选）	功能：“删除地图网格”菜单项点击事件响应函数

```
private void 删除地图网格ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ICommand command = new CmdDelMapGrid();
    command.OnCreate(m_PageLayoutControl.Object);
    command.OnClick();
}
```

（7）编译并运行程序，添加地图文档，切换视图为 PageLayout 地图，点击“创建地图网格”，再点击“删除地图网格”运行结果如图 9-12 所示。

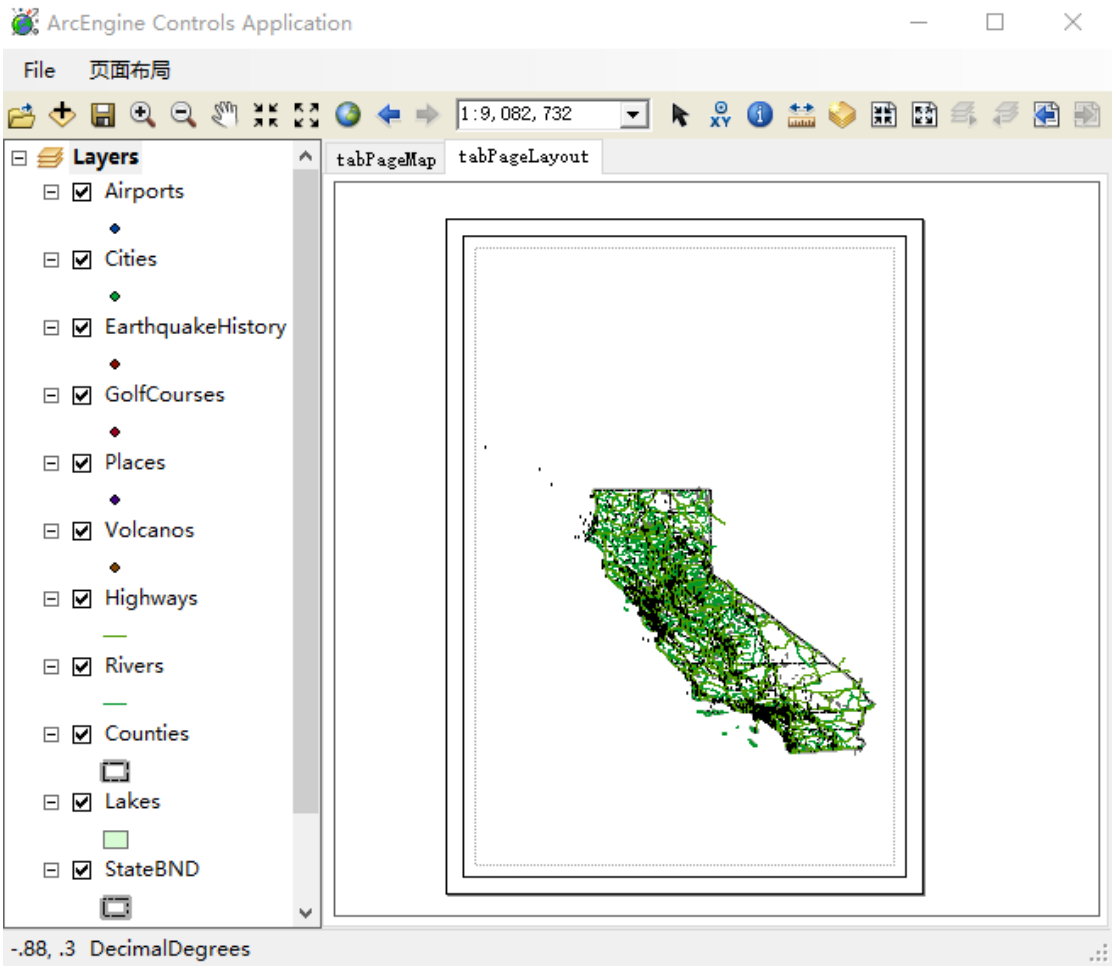

图 9-12　删除地图网格功能运行效果

9.5.3 设置地图框架属性

（1）为 Test9 项目添加三个新建项“Base Command”按钮命令，按钮命令

类分别命名为“CmdCreatBorder”、“CmdCreatShadow”和“CmdCreatBackG”，用来设置 MapFrame 的边框 Border、阴影 Shadow 和背景 Backgroud 属性。分别为按钮命令类“CmdCreatBorder”、“CmdCreatShadow”和“CmdCreatBackG”导入引用：

```
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Display;
```

（2）在“CmdCreatBorder”命令类中添加创建边框样式的函数代码，并在 OnClick 重载事件中调用，代码如下：

CmdCreatBorder.cs（节选）	功能：创建边框

```
private ISymbolBorder CreatSymbologyBorder()
{
    ISymbolBorder psimpleBorder = new SymbolBorderClass();
    ISimpleLineSymbol pLineSymbol = new SimpleLineSymbolClass();
    //设置样式
    pLineSymbol.Style = esriSimpleLineStyle.esriSLSSolid;
    pLineSymbol.Width = 5;
    //设置颜色
    IRgbColor pRgbColor = new RgbColorClass();
    pRgbColor.Red = 67 ;
    pRgbColor.Green = 205;
    pRgbColor.Blue = 128;
    pLineSymbol.Color = pRgbColor;
    psimpleBorder.LineSymbol = pLineSymbol;
    return psimpleBorder;
}
public override void OnClick()
{
    // TODO: Add CmdCreateBorder.OnClick implementation
    IGraphicsContainer graphicsContainer =
                                m_hookHelper.ActiveView.GraphicsContainer;
    IMapFrame mapFrame = (IMapFrame)graphicsContainer.FindFrame(
                                                    m_hookHelper.FocusMap);
    mapFrame.Border = CreatSymbologyBorder() as IBorder;
    m_hookHelper.ActiveView.Refresh();
}
```

（3）在“CmdCreatShadow”命令类中添加创建阴影样式的函数代码，并在 OnClick 重载事件中调用，代码如下：

CmdCreatShadow.cs（节选）	功能：创建阴影

```
private ISymbolShadow CreatSymbolShadow()
```

```
{
    ISymbolShadow pSymbolShadow = new SymbolShadowClass();
    ISimpleFillSymbol pSimpleFillSymbol = new SimpleFillSymbolClass();
    //设置样式
    pSimpleFillSymbol.Style = esriSimpleFillStyle.esriSFSSolid;
    //设置颜色
    IRgbColor pRgbColor = new RgbColorClass();
    pRgbColor.Red = 131;
    pRgbColor.Green =139 ;
    pRgbColor.Blue =139 ;
    pSimpleFillSymbol.Color = pRgbColor;
    pSymbolShadow.FillSymbol = pSimpleFillSymbol;
    return pSymbolShadow;
}
public override void OnClick()
{
    // TODO: Add CmdCreateShadow.OnClick implementation
    IGraphicsContainer graphicsContainer =
                                    m_hookHelper.ActiveView.GraphicsContainer;
    IFrameProperties pFrameProperties = (IFrameProperties)
                        graphicsContainer.FindFrame(m_hookHelper.FocusMap);
    pFrameProperties.Shadow = CreatSymbolShadow() as IShadow;
    m_hookHelper.ActiveView.Refresh();
}
```

（4）在“CmdCreatBackG”命令类中添加创建背景样式的函数代码，并在OnClick 重载事件中调用，代码如下：

CmdCreatShadow.cs（节选）	功能：创建背景

```
private ISymbolBackground CreatSymbologyBackGround()
{
    ISymbolBackground pSymbolBackground = new SymbolBackgroundClass();
    ISimpleFillSymbol pSimpleFillSymbol = new SimpleFillSymbolClass();
    //设置样式
    pSimpleFillSymbol.Style = esriSimpleFillStyle.esriSFSCross;
    //设置颜色
    IRgbColor pRgbColor = new RgbColorClass();
    pRgbColor.Red = 216;
    pRgbColor.Green = 191;
    pRgbColor.Blue = 216;
    pSimpleFillSymbol.Color = pRgbColor;
    pSymbolBackground.FillSymbol = pSimpleFillSymbol;
    return pSymbolBackground;
}
public override void OnClick()
```

```
{
    // TODO: Add CmdCreateBackG.OnClick implementation
    IGraphicsContainer graphicsContainer =
                                m_hookHelper.ActiveView.GraphicsContainer;
    IMapFrame mapFrame = (IMapFrame)graphicsContainer.FindFrame(
                                                    m_hookHelper.FocusMap);
    mapFrame.Background = CreatSymbologyBackGround() as IBackground;
    m_hookHelper.ActiveView.Refresh();
}
```

（5）在主窗体菜单 menuStrip1 控件的一级菜单“页面布局”下添加二级菜单为“地图框架属性”，在二级菜单项“地图框架属性”下添加三个三级菜单，为“设置边框”、“设置阴影”和“设置背景”，如图 9-13 所示。

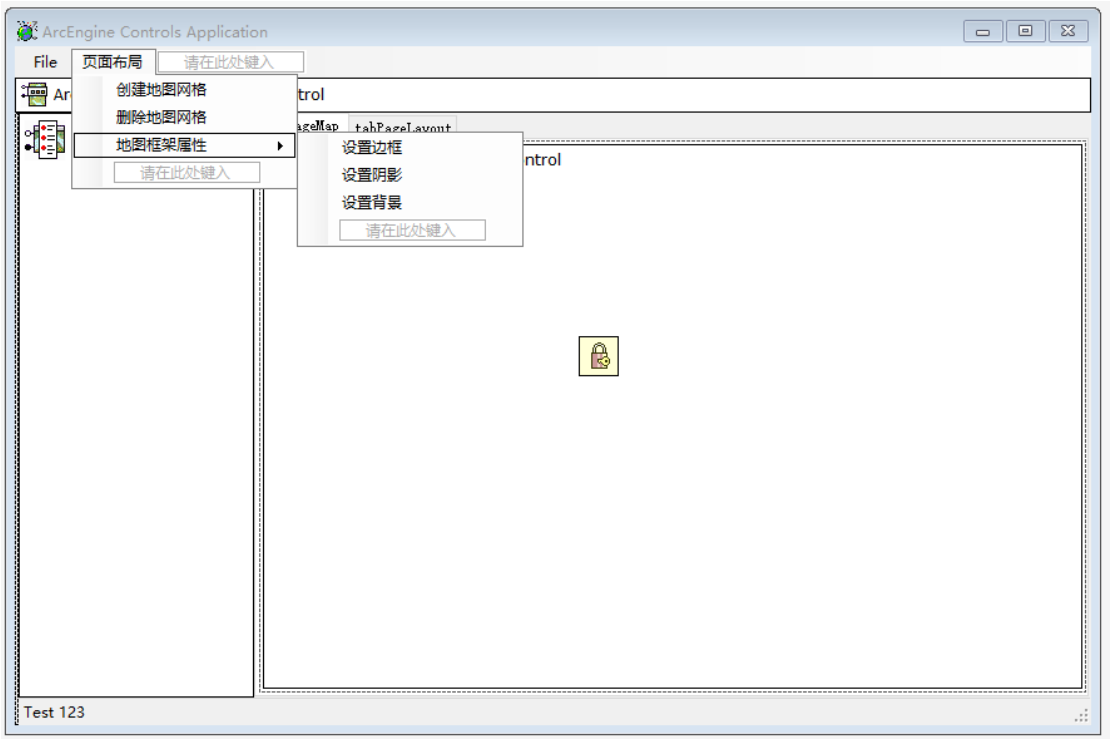

图 9-13　添加“地图框架属性”菜单项

双击主窗体菜单 menuStrip1 控件的“设置边框”、“设置阴影”和“设置背景”菜单项，为其添加点击事件响应函数，代码如下：

MainForm.cs（节选）	功能：“设置边框”、“设置阴影”和“设置背景”菜单项点击事件响应函数

```
private void  设置边框ToolStripMenuItem_Click(object sender, EventArgs e)
{
    ICommand command = new CmdCreateBorder();
    command.OnCreate(m_PageLayoutControl.Object);
    command.OnClick();
}
```

```
private void  设置阴影ToolStripMenuItem_Click(object sender, EventArgs e)
{
      ICommand command = new CmdCreateShadow();
      command.OnCreate(m_PageLayoutControl.Object);
      command.OnClick();
}
private void  设置背景ToolStripMenuItem_Click(object sender, EventArgs e)
{
      ICommand command = new CmdCreateBackG();
      command.OnCreate(m_PageLayoutControl.Object);
      command.OnClick();
}
```

（6）编译并运行程序，添加地图文档，切换视图为 PageLayout 地图，依次点击“设置边框”、“设置阴影”和“设置背景”菜单项，运行结果如图 9-14 所示。

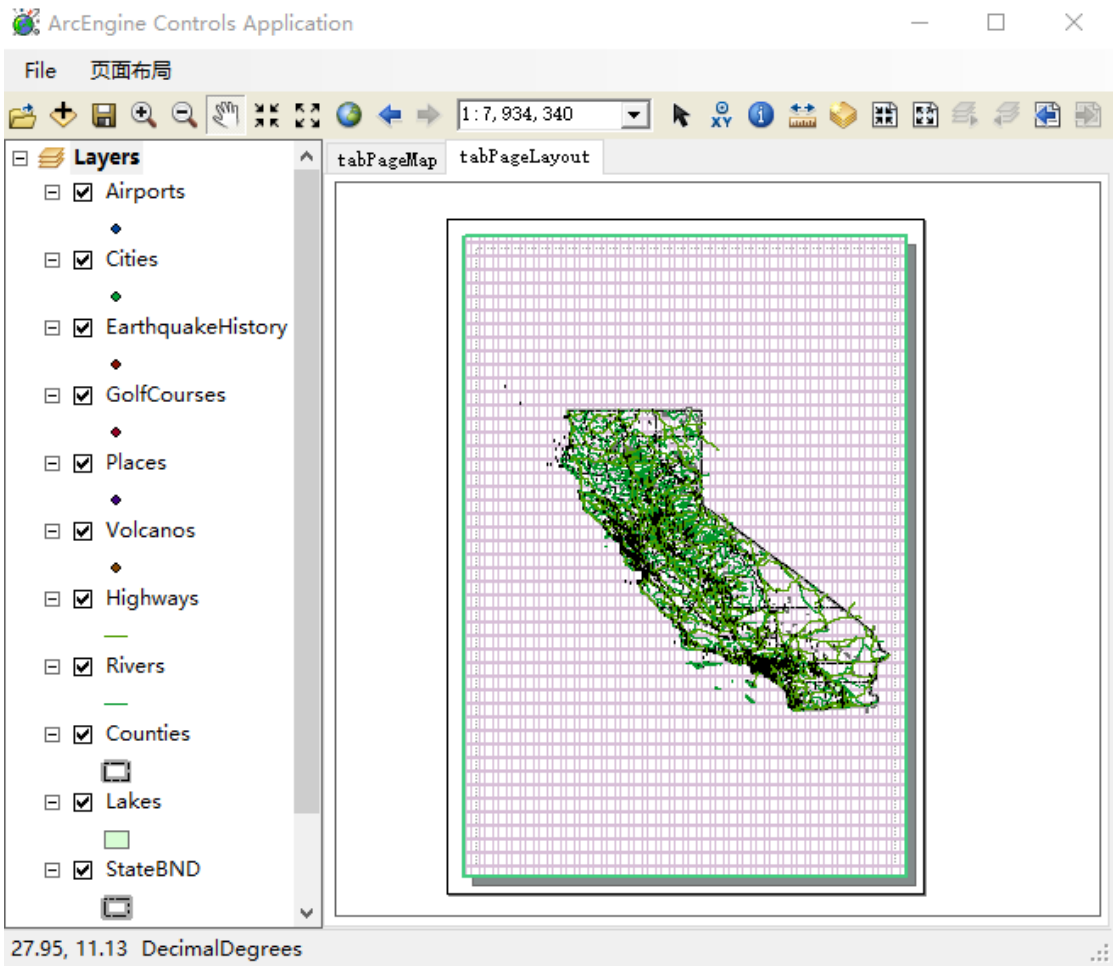

图 9-14　设置地图框架属性功能运行效果

9.5.4 地图整饰对象操纵

下面实现添加图例 Legend 对象、指北针 MarkerNorthArrow 对象、比例尺条 ScaleBar 对象、比例文本 ScaleText 对象等地图整饰对象。

（1）为 Test9 项目添加新建项“Base Tool”工具，工具类命名为“ToolMapSurround”。选择项目 Test9，右键点击【添加】→【新建项】，具体操作如图 9-15 所示。

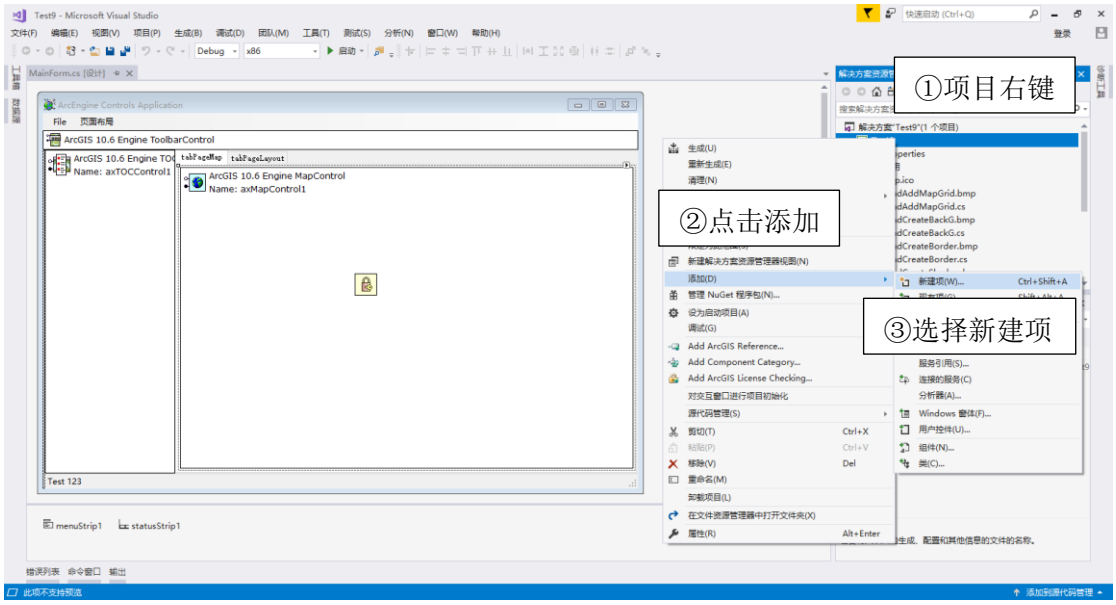

图 9-15　添加新建项

选择【ArcGIS】→【Extending ArcObjects】→【Base Tool】，具体操作如图 9-16 所示。

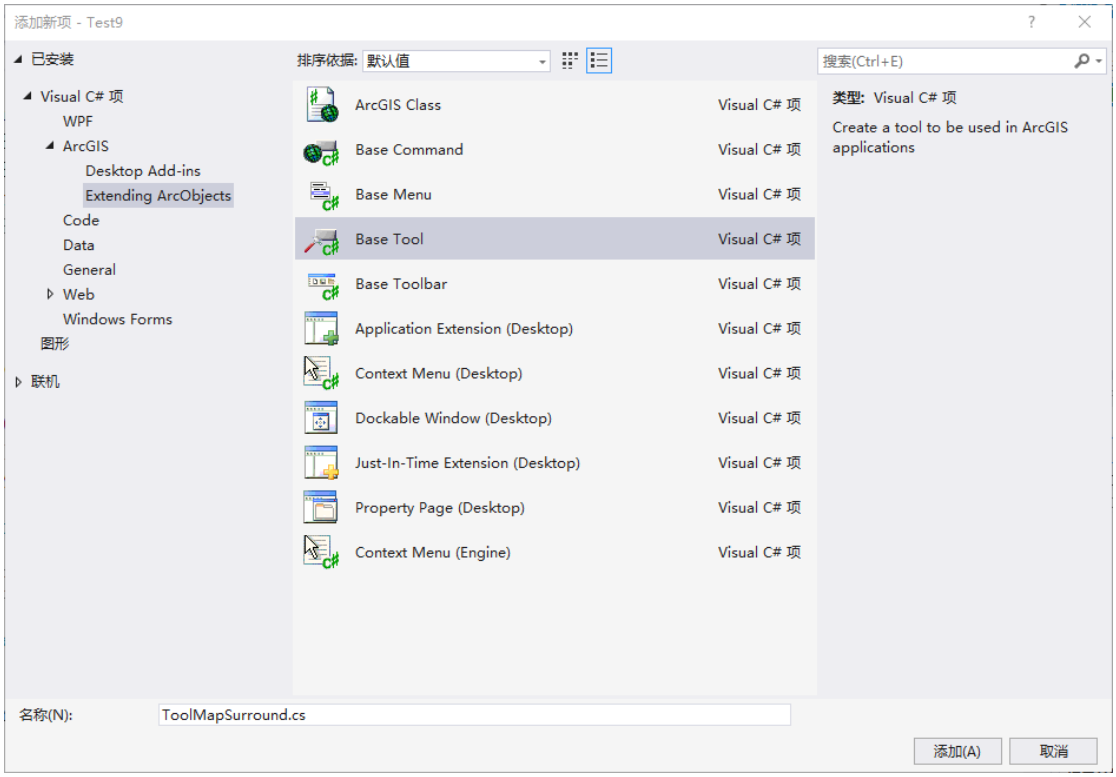

图 9-16　添加“Base Tool”工具类

（2）为工具类“ToolMapSurround”导入引用：

```
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.esriSystem;
using ESRI.ArcGIS.Display;
```

添加类“ToolMapSurround”的私有成员变量：

```
private string m_type = null;
```

并修改类构造函数，代码如下：

ToolMapSurround.cs（节选）	功能：修改构造函数

```
public ToolMapSurround(string type)
{
    m_type = type;
    // TODO: Define values for the public properties
    base.m_category = "";
    //localizable text
    base.m_caption = "";
    //localizable text
    base.m_message = "This should work in " +
                     "ArcMap/MapControl/PageLayoutControl";
    //localizable text
    base.m_toolTip = "";
```

```
        //localizable text
        base.m_name = "";
        //unique id, non-localizable (e.g. "MyCategory_MyTool")
        try
        {
            // TODO: change resource name if necessary
            string bitmapResourceName = GetType().Name + ".bmp";
            base.m_bitmap = new Bitmap(GetType(), bitmapResourceName);
            base.m_cursor = new System.Windows.Forms.Cursor(GetType(),
                                                    GetType().Name + ".cur");
        }
        catch (Exception ex)
        {
            System.Diagnostics.Trace.WriteLine(ex.Message, "Invalid Bitmap");
        }
}
```

（3）添加创建地图整饰对象成员函数，代码如下所示：

ToolMapSurround.cs（节选）	功能：添加创建地图整饰对象成员函数

```
private IMapSurround CreateMapSurround(UID ipUID, IEnvelope ipEnv,
                                        string sName, IMapSurround ipStyle)
{
    IMapSurround ms = null;
    IMapFrame mf = m_hookHelper.ActiveView.GraphicsContainer.FindFrame(
                        m_hookHelper.ActiveView.FocusMap) as IMapFrame;
    IMapSurroundFrame msf = mf.CreateSurroundFrame(ipUID, ipStyle);
    IElement ele = msf as IElement;
    ele.Geometry = ipEnv;
    ms = msf.MapSurround;
    ms.Name = sName;
    m_hookHelper.ActiveView.GraphicsContainer.AddElement(ele, 0);
    m_hookHelper.ActiveView.PartialRefresh(
                            esriViewDrawPhase.esriViewGraphics, null, null);
    return ms;
}
```

（4）在鼠标按下事件中创建地图整饰对象，代码如下：

ToolMapSurround.cs（节选）	功能：OnMouseDown 重载事件

```
public override void OnMouseDown(int Button, int Shift, int X, int Y)
{
    // TODO:  Add ToolMapSurround.OnMouseDown implementation
    string sName = "";
    UID uid = new UIDClass();
    uid.Value = m_type;
```

```
        switch (m_type)
        {
            case "esriCarto.Legend":
                sName = "图例";
                break;
            case "esriCarto.MarkerNorthArrow":
                sName = "指北针";
                break;
            case "esriCarto.ScaleLine":
                sName = "比例尺";
                break;
            case "esriCarto.ScaleText":
                sName = "比例尺";
                break;
            default:
                return;
        }
        IRubberBand ipRubber = new RubberEnvelopeClass();
        IEnvelope env = ipRubber.TrackNew(
                    m_hookHelper.ActiveView.ScreenDisplay, null) as IEnvelope;
        CreateMapSurround(uid, env, sName, null);
    }
```

（5）在主窗体菜单 menuStrip1 控件上一级菜单“页面布局”下添加二级菜单为“地图整饰对象”，为“地图整饰对象”二级菜单项下添加三级菜单为 ComboBox 控件，如图 9-17 所示。

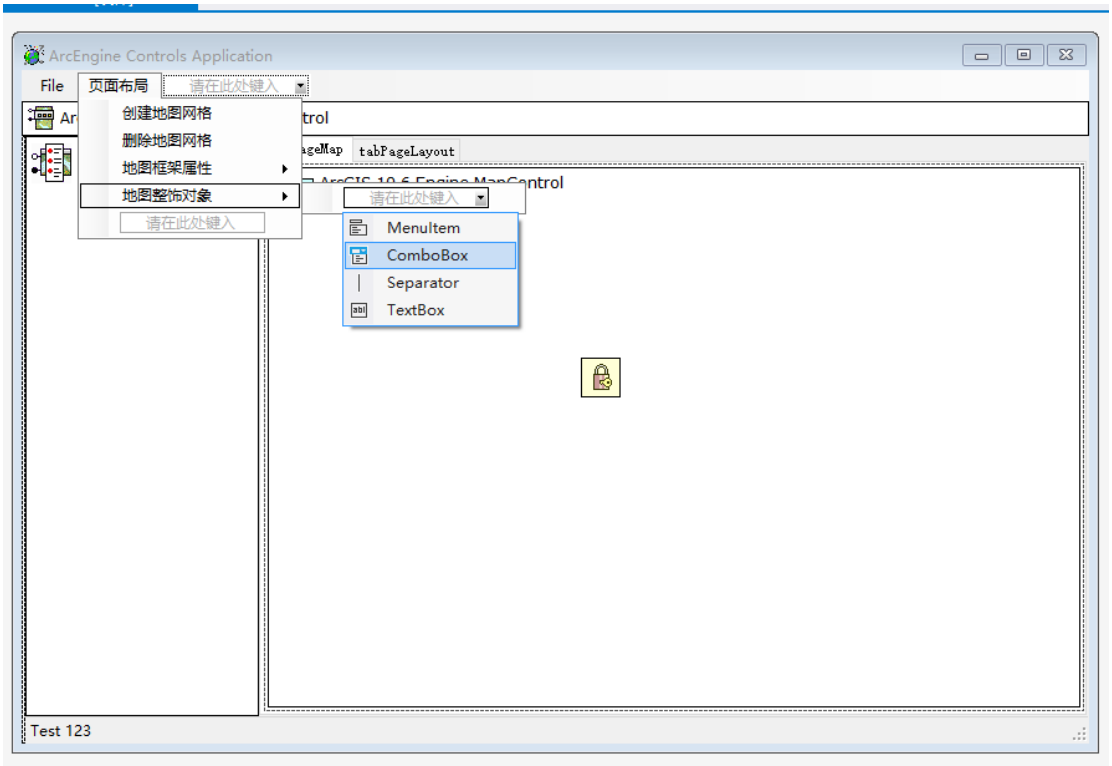

图 9-17　添加“ComboBox”三级菜单项

添加之后选中“toolStripComboBox1”三级菜单项，设置其 Items（集合）属性，打开“字符串集合编辑器”窗体，如图 9-18 所示。

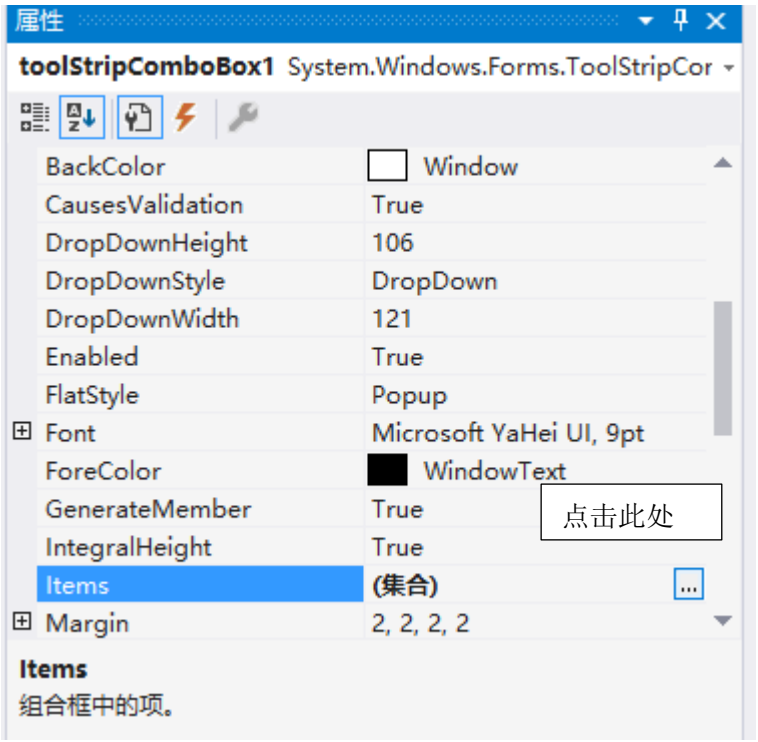

图 9-18　打开“字符串集合编辑器”窗体

设置 toolStripComboBox1 的集合，如图 9-19 所示。

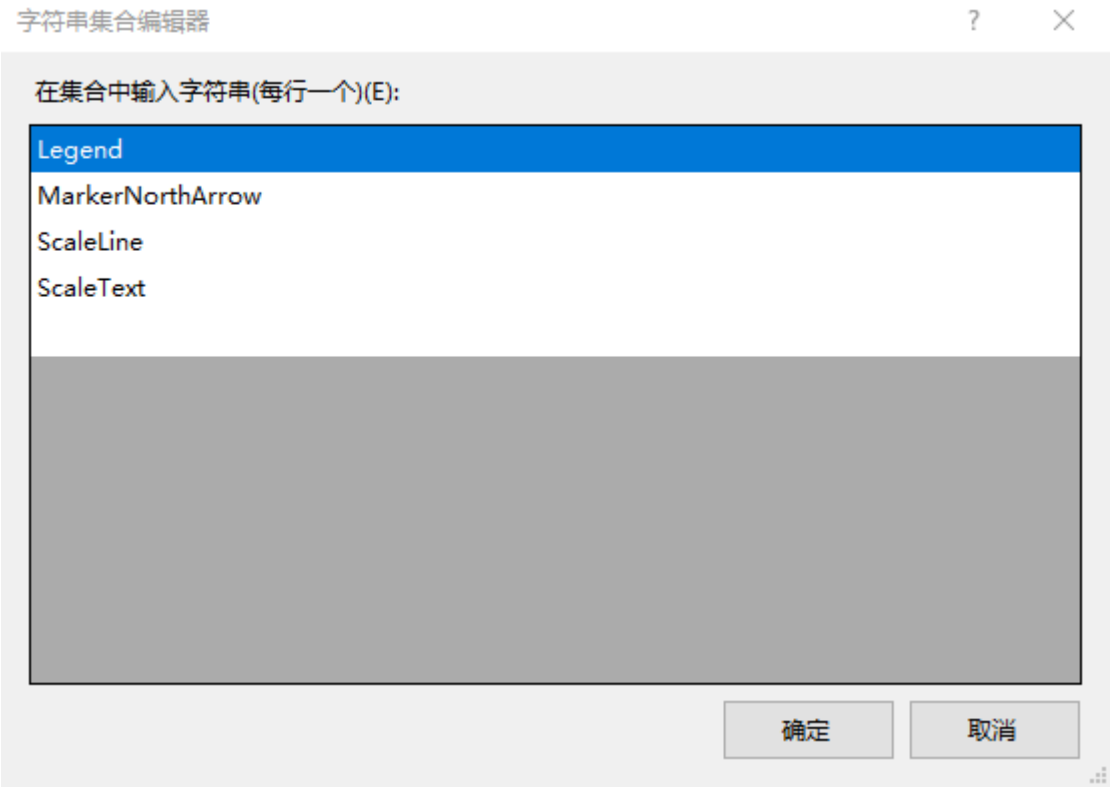

图 9-19　设置 toolStripComboBox1 的 Items 集合

（6）在组合框选择文本改变时，使用该工具，为组合框 toolStripComboBox1 控件的添加 SelectedIndexChanged 事件响应函数代码，代码如下：

MainForm.cs（节选）	功能：组合框内容改变事件

```
private void toolStripComboBox1_SelectedIndexChanged(object sender,
                                                     EventArgs e)
{
    string type = "esriCarto." + toolStripComboBox1.ComboBox.Text;
    if (tabControl1.SelectedIndex == 1)
    {
        ICommand command = new ToolMapSurround(type);
        command.OnCreate(m_PageLayoutControl.Object);
        command.OnClick();
        axPageLayoutControl1.CurrentTool = command as ITool;
    }
}
```

（7）编译并运行程序，程序运行结果如图 9-20 所示。

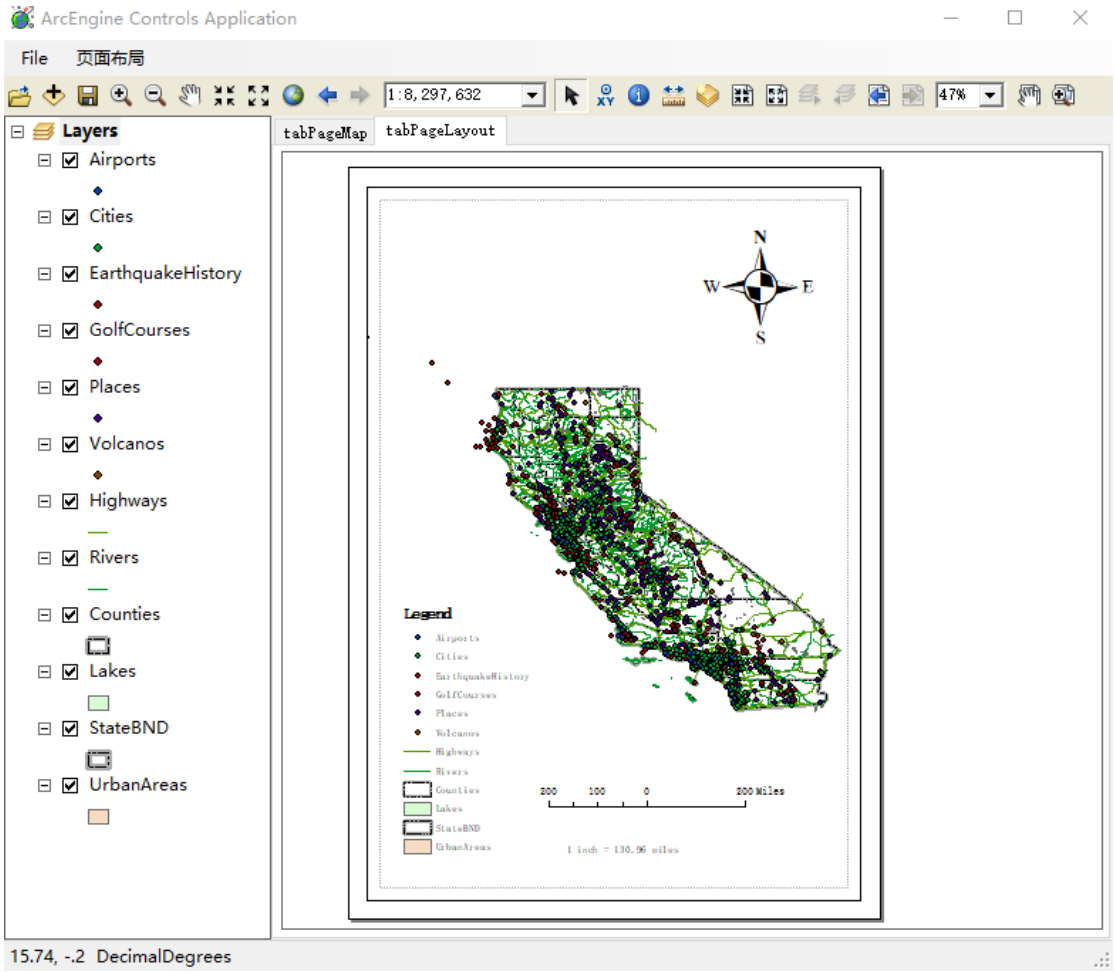

图 9-20　设置地图整饰功能运行效果