地理信息系统应用程序设计与开发

第五章 使用AO控件编程

主讲：张宝一
Email：zhangbaoyi.csu@qq.com

教学目标

- 熟悉MapControl、TOCContol、ToolbarControl
- 掌握Map对象的应用
- 熟悉AO控件开应用GIS的基本方法

教学重点和难点

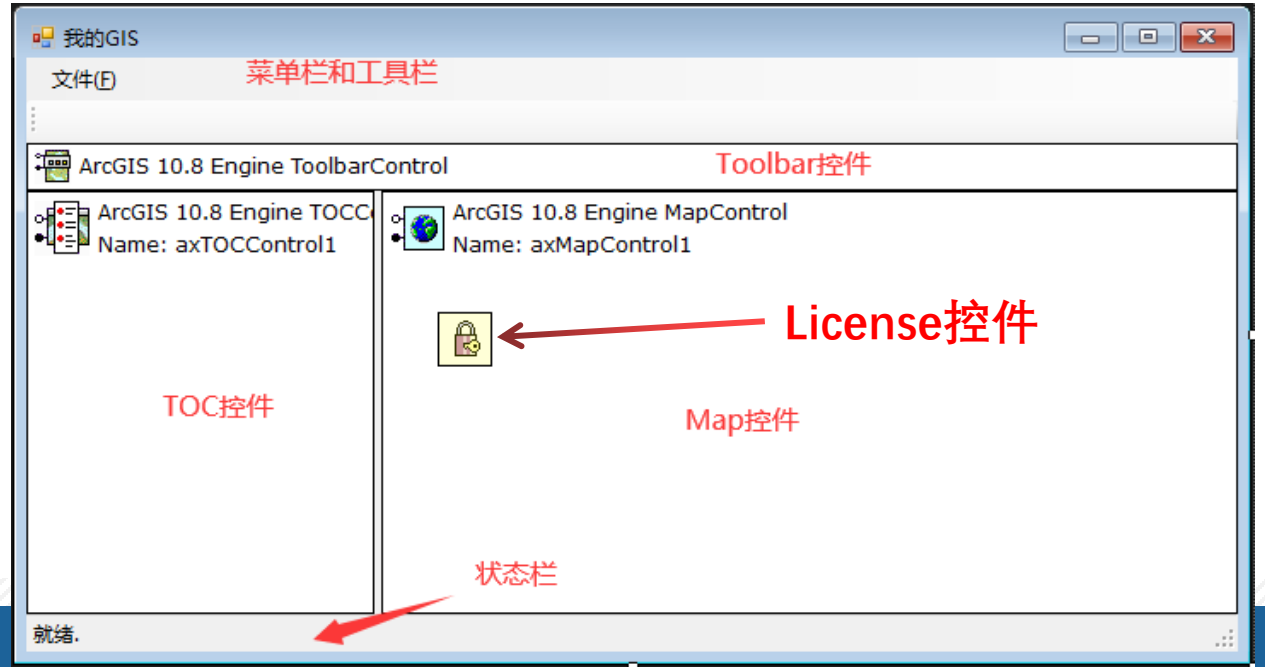
- Map控件、TOC控件常用操作

教学内容

- 5.1 MapControl控件
- 5.2 Map对象
- 5.3 TOCControl控件
- 5.4 ToolbarControl控件
- 5.5 开发实例

5.1 MapControl控件

- 第一个GIS桌面应用系统说起
  - 创建MapControl Application 项目
    - 0代码设计GIS程序
    - 集成MapControl、ToolbarControl、TOCControl
- 创建Windows Application(Engine) 项目



5.1 MapControl控件

- MapControl实现的主要接口有：
  - IMapControlDefault：地图控件缺省接口
  - IMapControl2：任何与MapControl相关的任务的出发点
  - IMapControl3：继承IMapControl2
  - IMapControlEvents2：事件接口
- 示例
  - IMapControlDefault pMapControl;
  - pMapControl = axMapControl1.Object as IMapControlDefault;
  - IMapControl2 pMapControl;
  - pMapControl = axMapControl1.Object as IMapControl2;
  - IMapControl3 pMapControl;
  - pMapControl = axMapControl1.Object as IMapControl3;

5.1 MapControl控件

MapControl常用事件

Name	Description
OnAfterDraw	Fires after the Map draws a specified view phase.
OnAfterScreenDraw	Fires after the Map contained by the MapControl has finished drawing.
OnBeforeScreenDraw	Fires before the Map contained by the MapControl starts to draw.
OnDoubleClick	Fires when the user presses and releases the mouse button twice in quick succession.
OnExtentUpdated	Fires after the extent (visible bounds) of the MapControl is changed.
OnFullExtentUpdated	Fires after the full extent (bounds) of the MapControl has changed.
OnKeyDown	Fires after a key is pressed on the keyboard.
OnKeyUp	Fires after a pressed key is released.
OnMouseDown	Fires when the user presses any mouse button while over the MapControl.
OnMouseMove	Fires when the user moves the mouse over the MapControl.
OnMouseUp	Fires when the user releases a mouse button while over the MapControl.
OnSelectionChanged	Fires when the current selection changes.
OnViewRefreshed	Fires when the view is refreshed before drawing occurs.

5.1 MapControl控件

地图文档操作

- m_mapControl = (IMapControl3)axMapControl1.Object;
- (1)新建地图文档

```
ICommand command = new NewMapDocument(); //自定义类，见下页
command.OnCreate(m_mapControl.Object);
command.OnClick();
```

- (2)打开文档

```
ICommand command = new ControlsOpenDocCommandClass();
command.OnCreate(m_mapControl.Object);
command.OnClick();
```

- (3)保存文档

```
ICommand command = new ControlsSaveAsDocCommandClass();
command.OnCreate(this.axMapControl1.Object);
command.OnClick();
```

NewMapDocument类:

```
public class NewMapDocument : BaseCommand
    {
        private IHookHelper m_hookHelper = null;
        //constructor
        public NewMapDocument()
        {
            //update the base properties
            base.m_category = ".NET Application";
            base.m_caption = "NewDocument";
            base.m_message = "Create a new map";
            base.m_toolTip = "Create a new map";
            base.m_name =
"DotNetTemplate_NewDocumentCommand";
        }
        /// <summary>
        /// Occurs when this command is created
        /// </summary>
        /// <param name="hook">Instance of the
application</param>
        public override void OnCreate(object hook)
        {
            if (m_hookHelper == null)
                m_hookHelper = new HookHelperClass();
```

```
        // 点击事件
        public override void OnClick()
        {
            IMapControl3 mapControl = (IMapControl3)m_hookHelper.Hook;
                result;
            IEngineEditor engineEditor = new EngineEditorClass();
            if ((engineEditor.EditState == esriEngineEditState.esriEngineStateEditing)
                && (engineEditor.HasEdits() == true))
            {
                DialogResult result = MessageBox.Show("是否保存当前地图？", "提示",
                    MessageBoxButtons.YesNoCancel,
                    MessageBoxIcon.Question);
                if (result == DialogResult.Cancel) return;
                else if (result == DialogResult.No)
                    engineEditor.StopEditing(false);
                else if (result == DialogResult.Yes)
                {
                    engineEditor.StopEditing(true);
                    //launch the save command
                    ICommand command = new ControlsSaveAsDocCommandClass();
                    command.OnCreate(m_hookHelper.Hook);
                    command.OnClick();
                }
            }
            //create a new Map
            IMap map = new MapClass();
            map.Name = "Map";
            //assign the new map to the MapControl
            mapControl.DocumentFilename = string.Empty;
            mapControl.Map = map;
        }
    }
```

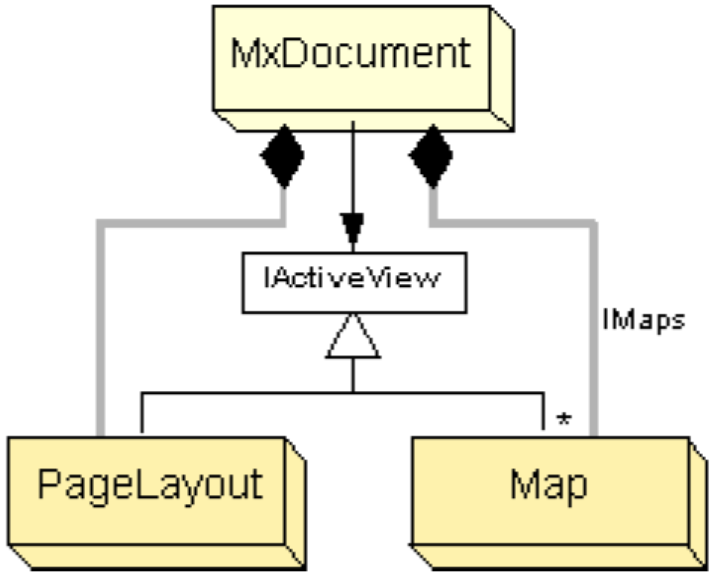
[示例] 在状态栏中显示鼠标移动时所在的地图坐标

```
private void axMapControl1_OnMouseDown(object sender,
                                       IMapControlEvents2_OnMouseDownEvent e)
{
    string coor = string.Format("x={0},y={1}", e.mapX, e.mapY);
    this.tslCoor.Text = coor;
}
```

5.2 Map和Layer类

(1)Map对象

- 数据的管理容器: 可以引入地理数据和可视化元素
- 数据的显示器: 它又可以让用户看到这些数据
- Map类实现的主要接口
  - IMap
    - 管理Map对象中的Layer对象、要素选择集、标注引擎和空间参考等
  - IActiveView
    - 管理地图视图
    - 方法Refresh、Draw等
  - IGraphicsContainer
    - 图元管理:AddElement、
    - DeleteElement
    - LocateElements等



5.2 Map和Layer类

Layer类

- Layer没有装载数据，只是获得数据的引用，是用于管理数据源的连接
- 在AO中地理数据始终是保存在GeoDatabase或者地理文件中
- 实现接口
  - ILayer
    - 常用属性和方法包括Name、Draw()、MaximumScale、MinimumScale、Visible等
  - ILayer2　　同上
  - IFeatureLayer
    - 除ILayer属性方法外，还有Search()、Selectable、DataSourceType、FeatureClass、Valid等
  - IRasterLayer

  - 更多属性和方法见AO Helper

5.2 Map和Layer类

FeatureLayer类

- 要素类图层，连接矢量数据和地图控件
- 实现接口
  - IFeatureLayer
    - 除ILayer属性方法外，还有Search()、Selectable、DataSourceType、FeatureClass、Valid等
  - ILayer、ILayerInfo、IDataLayer、IAttributeTable、IGeoDataSet、IFind、ITable
  - ITopologicalOperator
    - 方法包括Boundary(),Buffer(),Clip(),ConvexHull(),Cut(),Difference(),
    - Intersects(),IsSimple,Simplify(),SymmetricDifference(),Union() 等

5.2 Map和Layer对象

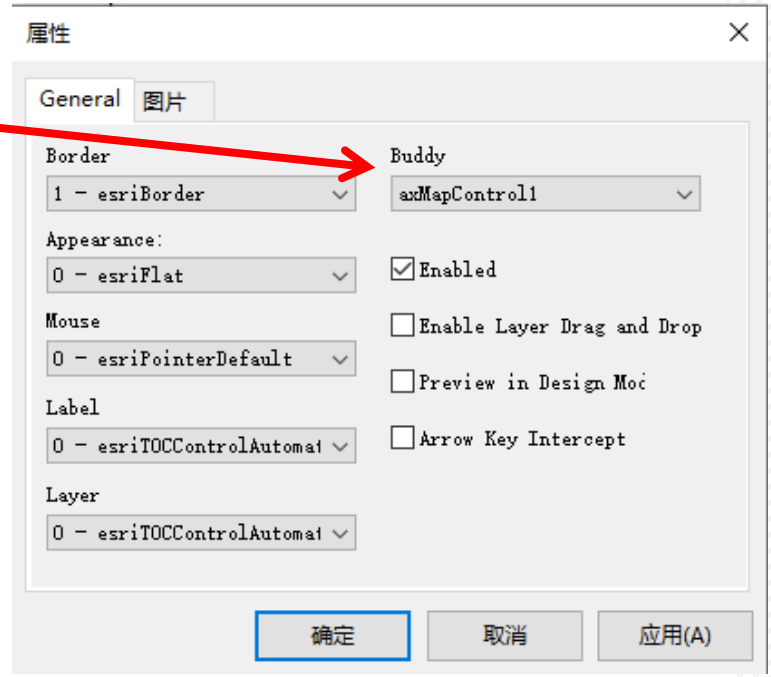
- 【示例】图层加载

```
① IWorkspace ws = null;
② IWorkspaceFactory wsf = new ShapefileWorkspaceFactory();
③ ws = wsf.OpenFromFile(@"d:\temp\csu", 0);
④ IFeatureWorkspace fws = (IFeatureWorkspace)ws;
⑤ IFeatureClass fc = fws.OpenFeatureClass("jmd.shp");
⑥ IFeatureLayer layer = new FeatureLayer();
⑦ layer.FeatureClass = fc;
⑧ layer.Name = fc.AliasName;
⑨ this.axMapControl1.AddLayer(layer);
```

- 还可使用MapControl的方法
  - AddLayerFromFile(string layerfile_path)
  - AddShapeFile(string folder,string shpfile)

TOCControl控件

- 是MapControl、PageLayoutControl、ReaderControl、SceneControl、GlobeControl等的“伙伴控件”
- 实现接口
  - ITOCBuddy,ITOCControl等
- 常用方法:
  - GetSelectedItem( ) :只能在OnMouseUp事件中使用
  - HitTest( ): 可以在OnMouseDown和OnMouseUp事件中使用
- 常用事件:
  - OnMouseDown,OnMouseMove,OnMouseUp,OnDoubleClick, OnBeginLabelEdit,OnEndLabelEdit, OnKeyDown, OnKeyUp等



5.3 TOCControl控件

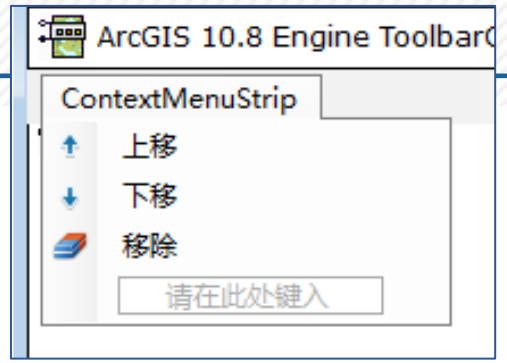
- [示例] 双击TOC的图层，获取当前选择的图层，判断是否为FeatureLayer

```
① private void axTOCControl1_OnDoubleClick(object sender,
  ITOCControlEvents_OnDoubleClickEvent e)
② {
③   esriTOCControlItem type = esriTOCControlItem.esriTOCControlItemNone;
④   IBasicMap basicMap = null;
⑤   ILayer layer = null;
⑥   object unk = null, data = null;
⑦   axTOCControl1.HitTest(e.x, e.y, ref type, ref basicMap,ref layer, ref unk, ref data);
⑧   if (type == esriTOCControlItem.esriTOCControlItemLayer
⑨      && layer != null
⑩      && e.button == 1)
⑪   {
⑫            MessageBox.Show(layer.Name);
⑬   }
⑭ }
```

5.3 TOCControl控件

- [示例]给TOC控件添加快捷菜单ContextMenu，添加菜单项：
  - 上移：将当前选中图层上移
  - 下移：将当前选中图层下移
  - 移除：将当前选中图层从地图中移除

5.3 TOCControl控件



- 【示例】
  - 设置快捷菜单
  - 为TOC添加OnMouseUp事件代码

```
① private void axTOCControl1_OnMouseUp(object sender,
  ITOCControlEvents_OnMouseUpEvent e)
② {
③    esriTOCControlItem type = esriTOCControlItem.esriTOCControlItemNone;
④    IBasicMap basicMap = null;
⑤    object unk = null, data = null;
⑥    axTOCControl1.GetSelectedItem(ref type, ref basicMap, ref m_selectedlayer, ref unk,
  ref data);
⑦    if (type == esriTOCControlItem.esriTOCControlItemLayer
⑧        && m_selectedlayer != null && e.button == 2)
⑨          contextMenuStrip1.Show(axTOCControl1, e.x, e.y);
⑩  }
```

- 【示例】上移菜单项点击事件代码

```
①  private void tsmUp_Click(object sender, EventArgs e)
②  {
③      int layerCount  = axMapControl1.LayerCount;
④      for (int i = 0; i < layerCount ; i++)
⑤      {
⑥          if (axMapControl1.get_Layer(i).Name == m_selectedlayer.Name
⑦              && i - 1 >= 0)
⑧          {
⑨              axMapControl1.MoveLayerTo(i, i - 1);
⑩              break;
⑪          }
⑫      }
⑬  }
```

- 【示例】下移菜单项点击事件代码

```
①  private void tsmDown_Click(object sender, EventArgs e)
②  {
③      int layerCount = axMapControl1.LayerCount;
④      for (int i = 0; i < layerCount ; i++)
⑤      {
⑥          if (axMapControl1.get_Layer(i).Name == m_selectedlayer.Name
⑦              && i + 1 < layerCount )
⑧           {
⑨              axMapControl1.MoveLayerTo(i, i + 1);
⑩              break;
⑪           }
⑫       }
⑬  }
```

【示例】移除菜单项点击事件代码

```
①   private void tsmRemove_Click(object sender, EventArgs e)
②   {
③       int index=-1;
④       for (int i = 0; i < this.axMapControl1.LayerCount; i++)
⑤         if (axMapControl1.get_Layer(i).Name == m_selectedlayer.Name)
⑥         {
⑦           index = i;
⑧           break;
⑨         }
⑩       if(index>=0)
⑪         this.axMapControl1.DeleteLayer(index);
⑫ }
```

5.4 ToolbarControl控件

- ToolbarControl控件
  - 和TOCControl控件一样，也是“伙伴控件”
  - 实现接口
    - IToolbarBuddy
  - 常用方法
    - AddItem( ) 用于添加自定义功能
  - 示例
    - ICommand command = new ControlsMapFullExtentCommandClass();
    - axToolbarControl1.AddItem(command,-1,-1,false,0,esriCommandStyles.esriCommandStyleIconOnly);
    - 注: 更多ICommand功能见AO Helper

5.5 应用示例

功能描述

- 添加自定义工具栏，在其中添加按钮：
  - （1）添加SHP：使用OpenFileDialog打开SHP文件，并加载到地图中
  - （2）画线：在地图绘制一条折线
  - （3）缓冲区分析：使用（2）绘制的折线，生成给定半径的缓冲区，选择与缓冲区相交的“居民地”
  - （4）浏览要素：浏览（3）选中的要素，使用表格显示属性列表
- 在状态栏中，添加坐标显示标签控件，用于显示当前鼠标移动时所在的坐标
- 为TOC控件添加快捷菜单，菜单项包括：
  - （1）上移：将选中的图层上移一层
  - （2）下移：将选中的图层下移一层
  - （3）移除：将选中的图层从地图中移除
  - （4）设置选择：将选中的图层设置为“要素可选”状态，其余层不可选

5.5 应用示例

- 图上画线:在地图控件的OnMouseDown事件中添加:

```
① private void axMapControl1_OnMouseDown(object sender,
  IMapControlEvents2_OnMouseDownEvent e)
② {          if(op==1)
③           {
④               IRubberBand rubber = new RubberLine();
⑤               IPolyline line = rubber.TrackNew(this.axMapControl1.ActiveView.ScreenDisplay, null)
  as IPolyline;
⑥               ISimpleLineSymbol pLineSym = new SimpleLineSymbol();
⑦               IRgbColor pColor = new RgbColor();
⑧               pColor.Red = 11;
⑨               pColor.Green = 120;
⑩               pColor.Blue = 233;
⑪               pLineSym.Color = pColor;
⑫               pLineSym.Style = esriSimpleLineStyle.esriSLSSolid;
⑬               pLineSym.Width = 2;
⑭               object symbol = pLineSym as object;
⑮               this.axMapControl1.DrawShape(line,ref symbol);
⑯               this.drawnLine = line;
⑰           }
```

画线

定义符号

5.5 应用示例

- 选择要素按钮点击事件：

```
① private void tsbSelect_Click(object sender, EventArgs e)
②         {
③             if (this.drawnLine == null)
④                 return;

⑤             ITopologicalOperator topo = (ITopologicalOperator)this.drawnLine;
⑥             IPolygon buffer = topo.Buffer(10) as IPolygon;

⑦             this.axMapControl1.Map.SelectByShape(buffer, null, false);
⑧             this.axMapControl1.ActiveView.Refresh();
⑨         }
```

5.5 应用示例

- 将选择的要素集转换为数据表DataTable部分代码：

```
① DataRow row;
② while(feature!=null)
③ {
④    row = dt.NewRow();
⑤    for (int i = 0; i < fields.FieldCount; i++)
⑥    {
⑦        if (fields.Field[i].Type == esriFieldType.esriFieldTypeGeometry)     row[i] = "Shape Object";
⑧        else     row[i] = feature.Value[i];
⑨    }
⑩  dt.Rows.Add(row);
⑪  feature = cur.NextRow() as IFeature;
⑫ }
```

插入要素数据

5.5 应用示例

- 将选择的要素集转换为数据表DataTable部分代码：

```
① if (this.axMapControl1.Map.SelectionCount == 0)
②                 return;
③  if (this.m_selectedlayer == null)
④                 return;
⑤ IFeatureSelection fs = m_selectedlayer as IFeatureSelection;
⑥ ICursor cur = null;
⑦ fs.SelectionSet.Refresh();
⑧ fs.SelectionSet.Search(null, true, out cur);
⑨ IFeature feature = cur.NextRow() as IFeature;
⑩ DataTable dt = new DataTable();
⑪ IFields fields = feature.Fields;
⑫ for (int i = 0; i < fields.FieldCount; i++)
⑬     dt.Columns.Add(fields.Field[i].Name);
```

创建数据表头

本章小结

- Map、TOC、Toolbar控件
- Map、Layer、FeatureLayer的应用
- 常用的地图事件及操作
- TOC事件、函数及操作