ArcGIS Engine+C#实例开发教程》第六讲 右键菜单添加与实现

时间：2009-04-08 21:42:39 来源：3SDN.Net 作者：3SDN 原创 点击量：4994

在上一讲中，我们完成了鹰眼功能，在这一讲中，大家将实现 TOCControl 控件和主地图控件的右键菜单。

在 AE 开发中，右键菜单有两种实现方式，一是使用 VS2005 自带的 ContextMenuStrip 控件，二是用 AE 封装的 IToolbarMenu 接口。相比较而言，后者更为简单实用，本文采用后者的实现方法。

1、创建右键菜单

在 Form1 类里面添加如下变量的定义：

```
//TOCControl 控件变量
private ITOCControl2 m_tocControl = null;

//TOCControl 中 Map 菜单

private IToolbarMenu m_menuMap = null;

//TOCControl 中图层菜单

private IToolbarMenu m_menuLayer = null;
```

在 Form1_Load 函数进行初始化，即菜单的创建：

```
m_menuMap = new ToolbarMenuClass();

m_menuLayer = new ToolbarMenuClass();
```

2、添加菜单项

第 1 步中创建的菜单可认为是菜单容器，里面什么都没有，具体的命令或工具作为菜单项添加到菜单容器才能工作。一般情况下，启动程序就要完成菜单项的添加，故此工作在 Form1_Load 函数完成。

当然，添加菜单项之前，必须实现相应命令或工具。这里的命令或工具可以 AE 内置的也可以是自定义的。AE 内置了许多可以直接调用的常用命令和工具，如 ControlsAddDataCommandClass，在 ESRI.ArcGIS.Controls 命名空间中，大家可以对象浏览器中查看。当然，这里也可以直接调用 AE 内置的菜单，如 ControlsFeatureSelectionMenu。另外，本讲也实现三自定义命令，以做示范。它们分别为图层可视控制命令（用于控制图层显示与否）、移除图层和放大到整个图层命令。实现方法也很简单，就是右击 3sdnMap 项目，选择“添加|类”，选择 C#普通的类模板，用以下代码覆盖系统自己生成的所有代码。

图层可视控制类 LayerVisibility 代码：

```
 using ESRI.ArcGIS.ADF.BaseClasses;
using ESRI.ArcGIS.Controls;

using ESRI.ArcGIS.Carto;

using ESRI.ArcGIS.SystemUI;



namespace _sdnMap

{

    /// <summary>

    ///  图层可视控制

    /// </summary>

        public sealed class LayerVisibility : BaseCommand, ICommandSubType

        {

                private IHookHelper m_hookHelper = new HookHelperClass();

                private long m_subType;



                public LayerVisibility()

                {

                }
```

```
        public override void OnClick()
        {
            for (int i=0; i <= m_hookHelper.FocusMap.LayerCount - 1; i++)
            {
                if (m_subType == 1)
m_hookHelper.FocusMap.get_Layer(i).Visible = true;
                if (m_subType == 2)
m_hookHelper.FocusMap.get_Layer(i).Visible = false;
            }

m_hookHelper.ActiveView.PartialRefresh(esriViewDrawPhase.esriViewGeography,null,null);
        }

        public override void OnCreate(object hook)
        {
            m_hookHelper.Hook = hook;
        }

        public int GetCount()
        {
            return 2;
        }
```

```
public void SetSubType(int SubType)
{
	m_subType = SubType;
}

public override string Caption
{
	get
	{
		if (m_subType == 1) return "Turn All Layers On";
		else	return "Turn All Layers Off";
	}
}

public override bool Enabled
{
	get
	{
		bool enabled = false; int i;
		if (m_subType == 1)
		{
```

```
                    for (i=0;i<=m_hookHelper.FocusMap.LayerCount - 1;i++)
                    {
                        if (m_hookHelper.ActiveView.FocusMap.get_Layer(i).Visible == false)
                        {
                            enabled = true;
                            break;
                        }
                    }
                }
                else
                {
                    for (i=0;i<=m_hookHelper.FocusMap.LayerCount - 1;i++)
                    {
                        if (m_hookHelper.ActiveView.FocusMap.get_Layer(i).Visible == true)
                        {
                            enabled = true;
                            break;
                        }
                    }
                }
```

```
                return enabled;
            }
        }
    }
}
```

移除图层类 RemoveLayer 代码：

```
using ESRI.ArcGIS.ADF.BaseClasses;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Controls;

namespace _sdnMap
{
    /// <summary>
    ///  删除图层
    /// </summary>
    public sealed class RemoveLayer : BaseCommand
    {
        private IMapControl3 m_mapControl;

        public RemoveLayer()
        {
            base.m_caption = "Remove Layer";
```

```
        }

        public override void OnClick()
        {
            ILayer layer = (ILayer)m_mapControl.CustomProperty;
            m_mapControl.Map.DeleteLayer(layer);
        }

        public override void OnCreate(object hook)
        {
            m_mapControl = (IMapControl3)hook;
        }

    }
}
```

放大至整个图层类 ZoomToLayer：

```
        using ESRI.ArcGIS.ADF.BaseClasses;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Controls;

namespace _sdnMap
{
```

```
/// <summary>
///  放大至整个图层
/// </summary>
public sealed class ZoomToLayer : BaseCommand
{
    private IMapControl3 m_mapControl;

    public ZoomToLayer()
    {
        base.m_caption = "Zoom To Layer";
    }

    public override void OnClick()
    {
        ILayer layer = (ILayer)m_mapControl.CustomProperty;
        m_mapControl.Extent = layer.AreaOfInterest;
    }

    public override void OnCreate(object hook)
    {
        m_mapControl = (IMapControl3)hook;
    }
```

```
    }

}
```

以上三个工具或命令的实现代码比较简单，在此不过多的分析，请读者自行理解。

下面在 Form1_Load 函数中进行菜单项的添加，代码如下：

```
        //添加自定义菜单项到 TOCCOntrol 的 Map 菜单中
//打开文档菜单

 m_menuMap.AddItem(new OpenNewMapDocument(m_controlsSynchronizer), -1, 0, false, esriCommandStyles.esriCommandStyleIconAndText);

//添加数据菜单

 m_menuMap.AddItem(new ControlsAddDataCommandClass(), -1, 1, false, esriCommandStyles.esriCommandStyleIconAndText);

 //打开全部图层菜单

 m_menuMap.AddItem(new LayerVisibility(), 1, 2, false, esriCommandStyles.esriCommandStyleTextOnly);

//关闭全部图层菜单

 m_menuMap.AddItem(new LayerVisibility(), 2, 3, false, esriCommandStyles.esriCommandStyleTextOnly);

 //以二级菜单的形式添加内置的“选择”菜单

 m_menuMap.AddSubMenu("esriControls.ControlsFeatureSelectionMenu", 4, true);

//以二级菜单的形式添加内置的“地图浏览”菜单

 m_menuMap.AddSubMenu("esriControls.ControlsMapViewMenu",5, true);



//添加自定义菜单项到 TOCCOntrol 的图层菜单中

 m_menuLayer = new ToolbarMenuClass();

//添加“移除图层”菜单项
```

m_menuLayer.AddItem(new RemoveLayer(), -1, 0, false, esriCommandStyles.esriCommandStyleTextOnly);

//添加“放大到整个图层”菜单项

m_menuLayer.AddItem(new ZoomToLayer(), -1, 1, true, esriCommandStyles.esriCommandStyleTextOnly);

//设置菜单的 Hook

m_menuLayer.SetHook(m_mapControl);

m_menuMap.SetHook(m_mapControl);

3、

弹出右键菜单

顾名思义，右键菜单是在鼠标右键按下的时候弹出，所以我们要添加 TOCControl1 控件的 OnMouseDown 事件，实现代码如下：

```
private void axTOCControl1_OnMouseDown(object sender, ITOCControlEvents_OnMouseDownEvent e)
{
    //如果不是右键按下直接返回
    if (e.button != 2) return;

    esriTOCControlItem item = esriTOCControlItem.esriTOCControlItemNone;
    IBasicMap map = null;
    ILayer layer = null;
    object other = null;
    object index = null;
```

```
        //判断所选菜单的类型
        m_tocControl.HitTest(e.x, e.y, ref item, ref map, ref layer, ref other, ref index);

        //确定选定的菜单类型，Map 或是图层菜单
        if (item == esriTOCControlItem.esriTOCControlItemMap)
            m_tocControl.SelectItem(map, null);
        else
            m_tocControl.SelectItem(layer, null);

        //设置 CustomProperty 为 layer (用于自定义的 Layer 命令)
        m_mapControl.CustomProperty = layer;

        //弹出右键菜单
        if (item == esriTOCControlItem.esriTOCControlItemMap)
            m_menuMap.PopupMenu(e.x, e.y, m_tocControl.hWnd);
        if (item == esriTOCControlItem.esriTOCControlItemLayer)
            m_menuLayer.PopupMenu(e.x, e.y, m_tocControl.hWnd);
}
```

同样的方法，我们也可以实现主地图控件的右键菜单，以方便地图浏览。添加 MapControl1 控件的 OnMouseDown 事件，实现代码如下：

```
        /// <summary>
/// 主地图控件的右键响应函数
/// </summary>
/// <param name="sender"></param>
/// <param name="e"></param>
private void axMapControl1_OnMouseDown(object sender, IMapControlEvents2_OnMouseDownEvent e)
{
    if (e.button == 2)
    {
        //弹出右键菜单
        m_menuMap.PopupMenu(e.x,e.y,m_mapControl.hWnd);
    }
}
```

4、编译运行

按 F5 编译运行程序，你会发现，原来右键菜单实现起来是这么的简单啊！

下一讲中，我将给大家带的是 TOCControl 控件中图层拖拽（Drag and Drop）的实现。敬请关注 3SDN.Net。