《ArcGIS Engine+C#实例开发教程》

第三讲 MapControl 与 PageLayoutControl 同步

时间：2009-04-03 21:17:02　来源：3SDN.Net　作者：3SDN 原创　点击量：13044

在 ArcMap 中，能够很方面地进行 MapView 和 Layout View 两种视图的切换，而且二者之间的数据是同步显示的。

关于两种视图同步的实现方法有多种，可以使用 ObjectCopy 对象进行数据硬拷贝，而比较简单的方法莫过于二者共享一份地图了，这也是最常用的方法。

1、新建同步类 ControlsSynchronizer

在解决方案面板中右击项目名，选择“添加|类”，在类别中选择“Visual C#项目项”，在模板中选择“类”，输入类名“ControlsSynchronizer.cs”，将以下代码覆盖自动生成的代码：

```
using System;
using System.Drawing;
using System.Collections;
using System.ComponentModel;
using System.Windows.Forms;
using System.IO;
using System.Runtime.InteropServices;
using ESRI.ArcGIS.esriSystem;
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Controls;
using ESRI.ArcGIS.SystemUI;

namespace _sdnMap
{
    /// <summary>
    /// This class is used to synchronize a gven PageLayoutControl and a MapControl.
    /// When initialized, the user must pass the reference of these control to the class, bind
    /// the control together by calling 'BindControls' which in turn sets a joined Map referenced
    /// by both control; and set all the buddy controls joined between these two controls.
    /// When alternating between the MapControl and PageLayoutControl, you should activate the visible control
    /// and deactivate the other by calling ActivateXXX.
    /// This calss is limited to a situation where the controls are not simultaneously visible.
    /// </summary>
    public class ControlsSynchronizer
    {
        #region class members
```

```
        private IMapControl3 m_mapControl = null;
        private IPageLayoutControl2 m_pageLayoutControl = null;
        private ITool m_mapActiveTool = null;
        private ITool m_pageLayoutActiveTool = null;
        private bool m_IsMapCtrlactive = true;

        private ArrayList m_frameworkControls = null;
        #endregion

        #region constructor

        /// <summary>
        /// 默认构造函数
        /// </summary>
        public ControlsSynchronizer()
        {
            //初始化 ArrayList
            m_frameworkControls = new ArrayList();
        }

        /// <summary>
        /// 构造函数
        /// </summary>
        /// <param name="mapControl"></param>
        /// <param name="pageLayoutControl"></param>
        public ControlsSynchronizer(IMapControl3 mapControl, IPageLayoutControl2
pageLayoutControl)
            : this()
        {
            //为类成员赋值
            m_mapControl = mapControl;
            m_pageLayoutControl = pageLayoutControl;
        }
        #endregion

        #region properties
        /// <summary>
        /// 取得或设置 MapControl
        /// </summary>
        public IMapControl3 MapControl
        {
            get { return m_mapControl; }
            set { m_mapControl = value; }
        }
```

```
        /// <summary>
        /// 取得或设置 PageLayoutControl
        /// </summary>
        public IPageLayoutControl2 PageLayoutControl
        {
            get { return m_pageLayoutControl; }
            set { m_pageLayoutControl = value; }
        }

        /// <summary>
        /// 取得当前 ActiveView 的类型
        /// </summary>
        public string ActiveViewType
        {
            get
            {
                if (m_IsMapCtrlactive)
                    return "MapControl";
                else
                    return "PageLayoutControl";
            }
        }

        /// <summary>
        /// 取得当前活动的 Control
        /// </summary>
        public object ActiveControl
        {
            get
            {
                if (m_mapControl == null || m_pageLayoutControl == null)
                    throw new Exception("ControlsSynchronizer::ActiveControl:\r\nEither
MapControl or PageLayoutControl are not initialized!");

                if (m_IsMapCtrlactive)
                    return m_mapControl.Object;
                else
                    return m_pageLayoutControl.Object;
            }
        }
        #endregion

        #region Methods
```

```
        /// <summary>
        ///  激活 MapControl 并解除 the PagleLayoutControl
        /// </summary>
        public void ActivateMap()
        {
            try
            {
                if (m_pageLayoutControl == null || m_mapControl == null)
                    throw new Exception("ControlsSynchronizer::ActivateMap:\r\nEither MapControl or PageLayoutControl are not initialized!");

                //缓存当前 PageLayout 的 CurrentTool
                if (m_pageLayoutControl.CurrentTool != null) m_pageLayoutActiveTool = m_pageLayoutControl.CurrentTool;

                //解除 PagleLayout
                m_pageLayoutControl.ActiveView.Deactivate();
                //激活 MapControl
                m_mapControl.ActiveView.Activate(m_mapControl.hWnd);

                //将之前 MapControl 最后使用的 tool，作为活动的 tool，赋给 MapControl 的 CurrentTool
                if (m_mapActiveTool != null) m_mapControl.CurrentTool = m_mapActiveTool;

                m_IsMapCtrlactive = true;

                //为每一个的 framework controls,设置 Buddy control 为 MapControl
                this.SetBuddies(m_mapControl.Object);
            }
            catch (Exception ex)
            {
                throw new Exception(string.Format("ControlsSynchronizer::ActivateMap:\r\n{0}", ex.Message));
            }
        }

        /// <summary>
        ///  激活 PagleLayoutControl 并减活 MapCotrol
        /// </summary>
        public void ActivatePageLayout()
        {
            try
            {
```

```
                if (m_pageLayoutControl == null || m_mapControl == null)
                    throw new Exception("ControlsSynchronizer::ActivatePageLayout:\r\nEither MapControl or PageLayoutControl are not initialized!");

                //缓存当前 MapControl 的 CurrentTool
                if (m_mapControl.CurrentTool != null) m_mapActiveTool = m_mapControl.CurrentTool;

                //解除 MapControl
                m_mapControl.ActiveView.Deactivate();

                //激活 PageLayoutControl
                m_pageLayoutControl.ActiveView.Activate(m_pageLayoutControl.hWnd);

                //将之前 PageLayoutControl 最后使用的 tool，作为活动的 tool，赋给 PageLayoutControl 的 CurrentTool
                if (m_pageLayoutActiveTool != null) m_pageLayoutControl.CurrentTool = m_pageLayoutActiveTool;

                m_IsMapCtrlactive = false;

                //为每一个的 framework controls,设置 Buddy control 为 PageLayoutControl
                this.SetBuddies(m_pageLayoutControl.Object);
            }
            catch (Exception ex)
            {
                throw new Exception(string.Format("ControlsSynchronizer::ActivatePageLayout:\r\n{0}", ex.Message));
            }
        }

        /// <summary>
        /// 给予一个地图, 置换 PageLayoutControl 和 MapControl 的 focus map
        /// </summary>
        /// <param name="newMap"></param>
        public void ReplaceMap(IMap newMap)
        {
            if (newMap == null)
                throw new Exception("ControlsSynchronizer::ReplaceMap:\r\nNew map for replacement is not initialized!");

            if (m_pageLayoutControl == null || m_mapControl == null)
                throw new Exception("ControlsSynchronizer::ReplaceMap:\r\nEither
```

```
MapControl or PageLayoutControl are not initialized!");

            //create a new instance of IMaps collection which is needed by the PageLayout
            //创建一个 PageLayout 需要用到的,新的 IMaps collection 的实例
            IMaps maps = new Maps();
            //add the new map to the Maps collection
            //把新的地图加到 Maps collection 里头去
            maps.Add(newMap);

            bool bIsMapActive = m_IsMapCtrlactive;

            //call replace map on the PageLayout in order to replace the focus map
            //we must call ActivatePageLayout, since it is the control we call 'ReplaceMaps'
            //调用 PageLayout 的 replace map 来置换 focus map
            //我们必须调用 ActivatePageLayout,因为它是那个我们可以调用"ReplaceMaps"
的 Control
            this.ActivatePageLayout();
            m_pageLayoutControl.PageLayout.ReplaceMaps(maps);

            //assign the new map to the MapControl
            //把新的地图赋给 MapControl
            m_mapControl.Map = newMap;

            //reset the active tools
            //重设 active tools
            m_pageLayoutActiveTool = null;
            m_mapActiveTool = null;

            //make sure that the last active control is activated
            //确认之前活动的 control 被激活
            if (bIsMapActive)
            {
                this.ActivateMap();
                m_mapControl.ActiveView.Refresh();
            }
            else
            {
                this.ActivatePageLayout();
                m_pageLayoutControl.ActiveView.Refresh();
            }
        }

        /// <summary>
        /// bind the MapControl and PageLayoutControl together by assigning a new joint focus
```

```
map
        /// 指定共同的 Map 来把 MapControl 和 PageLayoutControl 绑在一起
        /// </summary>
        /// <param name="mapControl"></param>
        /// <param name="pageLayoutControl"></param>
        /// <param name="activateMapFirst">true if the MapControl supposed to be activated first,如果 MapControl 被首先激活,则为 true</param>
        public void BindControls(IMapControl3 mapControl, IPageLayoutControl2 pageLayoutControl, bool activateMapFirst)
        {
            if (mapControl == null || pageLayoutControl == null)
                throw new Exception("ControlsSynchronizer::BindControls:\r\nEither MapControl or PageLayoutControl are not initialized!");
            m_mapControl = MapControl;
            m_pageLayoutControl = pageLayoutControl;

            this.BindControls(activateMapFirst);
        }

        /// <summary>
        /// bind the MapControl and PageLayoutControl together by assigning a new joint focus map
        /// 指定共同的 Map 来把 MapControl 和 PageLayoutControl 绑在一起
        /// </summary>
        /// <param name="activateMapFirst">true if the MapControl supposed to be activated first,如果 MapControl 被首先激活,则为 true</param>
        public void BindControls(bool activateMapFirst)
        {
            if (m_pageLayoutControl == null || m_mapControl == null)
                throw new Exception("ControlsSynchronizer::BindControls:\r\nEither MapControl or PageLayoutControl are not initialized!");

            //create a new instance of IMap
            //创造 IMap 的一个实例
            IMap newMap = new MapClass();
            newMap.Name = "Map";

            //create a new instance of IMaps collection which is needed by the PageLayout
            //创造一个新的 IMaps collection 的实例,这是 PageLayout 所需要的
            IMaps maps = new Maps();
            //add the new Map instance to the Maps collection
            //把新的 Map 实例赋给 Maps collection
            maps.Add(newMap);
```

```
            //call replace map on the PageLayout in order to replace the focus map
            //调用 PageLayout 的 replace map 来置换 focus map
            m_pageLayoutControl.PageLayout.ReplaceMaps(maps);
            //assign the new map to the MapControl
            //把新的 map 赋给 MapControl
            m_mapControl.Map = newMap;

            //reset the active tools
            //重设 active tools
            m_pageLayoutActiveTool = null;
            m_mapActiveTool = null;

            //make sure that the last active control is activated
            //确定最后活动的 control 被激活
            if (activateMapFirst)
                this.ActivateMap();
            else
                this.ActivatePageLayout();
        }

        /// <summary>
        ///by passing the application's toolbars and TOC to the synchronization class, it saves you the
        ///management of the buddy control each time the active control changes. This method ads the framework
        ///control to an array; once the active control changes, the class iterates through the array and
        ///calles SetBuddyControl on each of the stored framework control.
        /// </summary>
        /// <param name="control"></param>
        public void AddFrameworkControl(object control)
        {
            if (control == null)
                throw new Exception("ControlsSynchronizer::AddFrameworkControl:\r\nAdded control is not initialized!");

            m_frameworkControls.Add(control);
        }

        /// <summary>
        /// Remove a framework control from the managed list of controls
        /// </summary>
        /// <param name="control"></param>
        public void RemoveFrameworkControl(object control)
```

```
        {
            if (control == null)
                throw new Exception("ControlsSynchronizer::RemoveFrameworkControl:\r\nControl to be removed is not initialized!");

            m_frameworkControls.Remove(control);
        }

        /// <summary>
        /// Remove a framework control from the managed list of controls by specifying its index in the list
        /// </summary>
        /// <param name="index"></param>
        public void RemoveFrameworkControlAt(int index)
        {
            if (m_frameworkControls.Count < index)
                throw new Exception("ControlsSynchronizer::RemoveFrameworkControlAt:\r\nIndex is out of range!");

            m_frameworkControls.RemoveAt(index);
        }

        /// <summary>
        /// when the active control changes, the class iterates through the array of the framework controls
        ///    and calles SetBuddyControl on each of the controls.
        /// </summary>
        /// <param name="buddy">the active control</param>
        private void SetBuddies(object buddy)
        {
            try
            {
                if (buddy == null)
                    throw new Exception("ControlsSynchronizer::SetBuddies:\r\nTarget Buddy Control is not initialized!");

                foreach (object obj in m_frameworkControls)
                {
                    if (obj is IToolbarControl)
                    {
                        ((IToolbarControl)obj).SetBuddyControl(buddy);
                    }
                    else if (obj is ITOCControl)
```

```
                    {
                        ((ITOCControl)obj).SetBuddyControl(buddy);
                    }
                }
            }
            catch (Exception ex)
            {
                throw new Exception(string.Format("ControlsSynchronizer::SetBuddies:\r\n{0}", ex.Message));
            }
        }
        #endregion
    }
}
```

2、新建 Maps 类

在同步类中，要用到 Maps 类，用于管理地图对象。与新建同步类 ControlsSynchronizer 类似，我们新建一 Maps 类，其所有代码如下所示：

```
 using System;
using System.Collections;
using System.Collections.Generic;
using System.Text;
using System.Runtime.InteropServices;

using ESRI.ArcGIS.Carto;

namespace _sdnMap
{
    [Guid("f27d8789-fbbc-4801-be78-0e3cd8fff9d5")]
    [ClassInterface(ClassInterfaceType.None)]
    [ProgId("_sdnMap.Maps")]
    public class Maps : IMaps, IDisposable
    {
        //class member - using internally an ArrayList to manage the Maps collection
        private ArrayList m_array = null;

        #region class constructor
        public Maps()
        {
            m_array = new ArrayList();
        }
        #endregion
```

```
#region IDisposable Members

/// <summary>
/// Dispose the collection
/// </summary>
public void Dispose()
{
	if (m_array != null)
	{
		m_array.Clear();
		m_array = null;
	}
}

#endregion

#region IMaps Members

/// <summary>
/// Remove the Map at the given index
/// </summary>
/// <param name="Index"></param>
public void RemoveAt(int Index)
{
	if (Index > m_array.Count || Index < 0)
		throw new Exception("Maps::RemoveAt:\r\nIndex is out of range!");

	m_array.RemoveAt(Index);
}

/// <summary>
/// Reset the Maps array
/// </summary>
public void Reset()
{
	m_array.Clear();
}

/// <summary>
/// Get the number of Maps in the collection
/// </summary>
public int Count
{
	get
```

```
        {
            return m_array.Count;
        }
    }

    /// <summary>
    /// Return the Map at the given index
    /// </summary>
    /// <param name="Index"></param>
    /// <returns></returns>
    public IMap get_Item(int Index)
    {
        if (Index > m_array.Count || Index < 0)
            throw new Exception("Maps::get_Item:\r\nIndex is out of range!");

        return m_array[Index] as IMap;
    }

    /// <summary>
    /// Remove the instance of the given Map
    /// </summary>
    /// <param name="Map"></param>
    public void Remove(IMap Map)
    {
        m_array.Remove(Map);
    }

    /// <summary>
    /// Create a new Map, add it to the collection and return it to the caller
    /// </summary>
    /// <returns></returns>
    public IMap Create()
    {
        IMap newMap = new MapClass();
        m_array.Add(newMap);

        return newMap;
    }

    /// <summary>
    /// Add the given Map to the collection
    /// </summary>
    /// <param name="Map"></param>
    public void Add(IMap Map)
```

```
        {
            if (Map == null)
                throw new Exception("Maps::Add:\r\nNew Map is mot initialized!");

            m_array.Add(Map);
        }

        #endregion
    }
}
```

3、新建打开文档类 OpenNewMapDocument

由于从工具栏自带的打开按钮打开地图文档的时候，不会自动进行两种视图之间的同步，所以我们要自己派生一个 OpenNewMapDocument 类，用于打开地图文档。

右击项目名，选择“添加|类”，再选择 ArcGIS 类别中的 BaseCommand 模板，输入类名为“OpenNewMapDocument.cs”。

首先添加引用：

```
using System.Windows.Forms;
using ESRI.ArcGIS.Carto;
```

再添加如下成员变量：

```
 private ControlsSynchronizer m_controlsSynchronizer = null;
```

修改默认的构造函数如下所示：

```
 //添加参数
public OpenNewMapDocument(ControlsSynchronizer controlsSynchronizer)
        {
            //
            // TODO: Define values for the public properties
            //
//设定相关属性值
            base.m_category = "Generic"; //localizable text
            base.m_caption = "Open";    //localizable text
```

```
            base.m_message = "This should work in ArcMap/MapControl/PageLayoutControl";
//localizable text
            base.m_toolTip = "Open";  //localizable text
            base.m_name = "Generic_Open";       //unique id, non-localizable (e.g.
"MyCategory_MyCommand")

    //初始化 m_controlsSynchronizer
            m_controlsSynchronizer = controlsSynchronizer;
            try
            {
                //
                // TODO: change bitmap name if necessary
                //
                string bitmapResourceName = GetType().Name + ".bmp";
                base.m_bitmap = new Bitmap(GetType(), bitmapResourceName);
            }
            catch (Exception ex)
            {
                System.Diagnostics.Trace.WriteLine(ex.Message, "Invalid Bitmap");
            }
        }
```

再在 OnClick 函数中添加如下代码：

```
 public override void OnClick()
        {
            // TODO: Add OpenNewMapDocument.OnClick implementation
            OpenFileDialog dlg = new OpenFileDialog();
            dlg.Filter = "Map Documents (*.mxd)|*.mxd";
            dlg.Multiselect = false;
            dlg.Title = "Open Map Document";
            if (dlg.ShowDialog() == DialogResult.OK)
            {
                string docName = dlg.FileName;
                IMapDocument mapDoc = new MapDocumentClass();
                if                                          (mapDoc.get_IsPresent(docName)
&& !mapDoc.get_IsPasswordProtected(docName))
                {
                    mapDoc.Open(docName, string.Empty);
                    IMap map = mapDoc.get_Map(0);
                    m_controlsSynchronizer.ReplaceMap(map);
```

```
                mapDoc.Close();
            }
        }
    }
```

在添加类时，模板会自动添加一个名为“OpenNewMapDocument.bmp”的图标，你可以自己修改或者替换为打开的文件夹的图标。

4、两种视图的同步

在 3sdnMap.cs 中添加成员变量，即同步类对象：

```
 private ControlsSynchronizer m_controlsSynchronizer = null;
```

在 Form1_Load 函数中进行初始化工作：

```
 //初始化 controls synchronization calss
m_controlsSynchronizer = new
ControlsSynchronizer(m_mapControl, m_pageLayoutControl);
      //把 MapControl 和 PageLayoutControl 绑定起来(两个都指向同一个 Map),然后设置
MapControl 为活动的 Control
       m_controlsSynchronizer.BindControls(true);
      //为了在切换 MapControl 和 PageLayoutControl 视图同步，要添加 Framework Control
     m_controlsSynchronizer.AddFrameworkControl(axToolbarControl1.Object);
     m_controlsSynchronizer.AddFrameworkControl(this.axTOCControl1.Object);
// 添加打开命令按钮到工具条
OpenNewMapDocument openMapDoc = new
OpenNewMapDocument(m_controlsSynchronizer);
axToolbarControl1.AddItem(openMapDoc, -1, 0, false, -1,
esriCommandStyles.esriCommandStyleIconOnly);
```

因为我们自动派生了打开文档类，并自己将其添加到工具条，所以我们就不需要工具条原来的“打开”按钮了，可以 ToolbarControl 的属性中将其删除。
下面，我们可完成上一讲遗留的功能了。

```
 /// <summary>
```

```
        /// 新建地图命令
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void New_Click(object sender, EventArgs e)
        {
            //询问是否保存当前地图
            DialogResult res = MessageBox.Show("是否保存当前地图?", "提示",
MessageBoxButtons.YesNo, MessageBoxIcon.Question);
            if (res == DialogResult.Yes)
            {
                //如果要保存，调用另存为对话框
                ICommand command = new ControlsSaveAsDocCommandClass();
                if (m_mapControl != null)
                    command.OnCreate(m_controlsSynchronizer.MapControl.Object);
                else

command.OnCreate(m_controlsSynchronizer.PageLayoutControl.Object);
                command.OnClick();
            }
      //创建新的地图实例
            IMap map = new MapClass();
            map.Name = "Map";
            m_controlsSynchronizer.MapControl.DocumentFilename = string.Empty;
            //更新新建地图实例的共享地图文档
            m_controlsSynchronizer.ReplaceMap(map);
        }
/// <summary>
        /// 打开地图文档 Mxd 命令
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void Open_Click(object sender, EventArgs e)
        {
if (this.axMapControl1.LayerCount > 0)
            {
                DialogResult result = MessageBox.Show("是否保存当前地图？", "警告",
MessageBoxButtons.YesNoCancel, MessageBoxIcon.Question);
                if (result == DialogResult.Cancel) return;
                if (result == DialogResult.Yes) this.Save_Click(null, null);
            }
            OpenNewMapDocument openMapDoc =
new OpenNewMapDocument(m_controlsSynchronizer);
            openMapDoc.OnCreate(m_controlsSynchronizer.MapControl.Object);
```

```
            openMapDoc.OnClick();
}
```

在添加数据 AddData 时，我们也要进行地图共享，故在 AddData_Click 函数后面添加如下代码：

```
 IMap pMap = this.axMapControl1.Map;
this.m_controlsSynchronizer.ReplaceMap(pMap);
```

在另存为地图文档时，有可能会丢失数据，因此我们需要提示用户以确认操作，故需修改 SaveAs_Click 函数，如下所示：

```
 /// <summary>
        /// 另存为地图文档命令
        /// </summary>
        /// <param name="sender"></param>
        /// <param name="e"></param>
        private void SaveAs_Click(object sender, EventArgs e)
        {
            //如果当前视图为 MapControl 时，提示用户另存为操作将丢失 PageLayoutControl 中的设置
            if (m_controlsSynchronizer.ActiveControl is IMapControl3)
            {
                if (MessageBox.Show("另存为地图文档将丢失制版视图的设置\r\n 您要继续吗?", "提示", MessageBoxButtons.YesNo, MessageBoxIcon.Question) == DialogResult.No)
                    return;
            }
            //调用另存为命令
            ICommand command = new ControlsSaveAsDocCommandClass();
            command.OnCreate(m_controlsSynchronizer.ActiveControl);
            command.OnClick();
        }
```

在切换视图时，我们要激活相关的视图，故在设计视图的属性面板中选择 tabControl2 控件，再选择事件按钮，找到“SelectedIndexChanged”事件双击添加之。其实现代码如下所示：

```
 /// <summary>
        /// 切换地图和制版视图
        /// </summary>
```

```
/// <param name="sender"></param>
/// <param name="e"></param>
private void tabControl2_SelectedIndexChanged(object sender, EventArgs e)
{
    if (this.tabControl2.SelectedIndex == 0)
    {
        //激活 MapControl
        m_controlsSynchronizer.ActivateMap();
    }
    else
    {
        //激活 PageLayoutControl
        m_controlsSynchronizer.ActivatePageLayout();
    }
}
```

5、编译运行

按 F5 编译运行程序，至此我们完成了 MapControl 和 PageLayoutControl 两种视图的同步工作。

在下一讲中，我将给大家带来的是状态栏的相关操作，敬请关注 3SDN.NET。