用户名: 密码:
登陆 注册
加入收藏 | 设为首页 | 会员中心 | 我要投稿 | RSS



《ArcGIS Engine+C#实例开发教程》第八讲 属性数据表的查询显示

时间：2009-05-01 06:31:26 来源：3SDN.Net 作者：3SDN原创 点击量： 5515

在上一讲中，我们完成了图层符号选择器的制作。这一讲中，我们将实现图层属性数据表的查询显示。

在ArcMap中，单击图层右键菜单中的“Open Attribute Table”命令，便可弹出属性数据表。本讲将完成类似的功能，效果如下：



FID	Shape	FNODE_	TNODE_
0	Polyline	258	259
1	Polyline	260	261
2	Polyline	262	263
3	Polyline	264	264
4	Polyline	265	266
5	Polyline	267	268
6	Polyline	269	270
7	Polyline	271	270
8	Polyline	265	272
9	Polyline	273	274
10	Polyline	275	276
11	Polyline	277	278
12	Polyline	276	279

图1

数据表显示，我们用了DataGridView控件。DataGridView 控件提供一种强大而灵活的以表格形式显示数据的方式。可以使用 DataGridView 控件来显示少量数据的只读视图，也可以对其进行缩放以显示特大数据集的可编辑视图。我们可以很方便地把一个DataTable作为数据源绑定到DataGridView控件中。

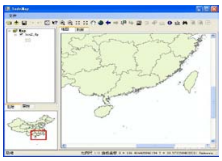
本讲的思路大体如下：首先根据图层属性中的字段创建一个空的DataTable，然后根据数据内容一行行填充DataTable数据，再将DataTable绑定到DataGridView控件，最后调用并显示属性表窗体。

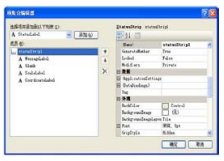
1.创建属性表窗体

推荐资讯


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实

相关文章

- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》教程Bug及优化方案1
- 《ArcGIS Engine+C#实例开发教程》第六讲 右键菜单添加
- 《ArcGIS Engine+C#实例开发教程》第五讲 鹰眼的实现
- 《ArcGIS Engine+C#实例开发教程》第四讲 状态栏信息的
- 《ArcGIS Engine+C#实例开发教程》第三讲 MapControl与
- 《ArcGIS Engine+C#实例开发教程》第二讲 菜单的添加及
- 《ArcGIS Engine+C#实例开发教程》第一讲 桌面GIS应用
- AO开发客户化组件（DLL+ArcMap）入门实例

栏目更新

- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发

新建一个Windows窗体，命名为“AttributeTableFrm.cs”。

从工具箱拖一个DataGridView控件到窗体，并将其Dock属性设置为“Fill”。

添加如下引用：

```
using ESRI.ArcGIS.Carto;
using ESRI.ArcGIS.Controls;
using ESRI.ArcGIS.esriSystem;
using ESRI.ArcGIS.SystemUI;
using ESRI.ArcGIS.Geometry;
using ESRI.ArcGIS.Geodatabase;
```

2.创建空DataTable

首先传入ILayer，再查询到ITable，从ITable中的Fileds中获得每个Field，再根据Filed设置DataTable的DataColumn，由此创建一个只含图层字段的空DataTable。实现函数如下：

```
/// <summary>
/// 根据图层字段创建一个只含字段的空DataTable
/// </summary>
/// <param name="pLayer"></param>
/// <param name="tableName"></param>
/// <returns></returns>
private static DataTable CreateDataTableByLayer(ILayer pLayer, string tableName)
{
//创建一个DataTable表
DataTable pDataTable = new DataTable(tableName);
//取得ITable接口
ITable pTable = pLayer as ITable;
IField pField = null;
DataColumn pDataColumn;
//根据每个字段的属性建立DataColumn对象
for (int i = 0; i < pTable.Fields.FieldCount; i++)
{
pField = pTable.Fields.get_Field(i);
//新建一个DataColumn并设置其属性
pDataColumn = new DataColumn(pField.Name);
if (pField.Name == pTable.OIDFieldName)
```


- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- （连载三）VC、OpenGL、ArcG
- （连载二）VC、OpenGL、ArcG
- （连载一）VC、OpenGL、ArcG

栏目热门

- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- ArcGIS Engine中的数据访问
- （连载一）VC、OpenGL、ArcG
- ArcGIS Engine的license部署

```
{
pDataColumn.Unique = true;//字段值是否唯一
}
//字段值是否允许为空
pDataColumn.AllowDBNull = pField.IsNullable;
//字段别名
pDataColumn.Caption = pField.AliasName;
//字段数据类型
pDataColumn.DataType = System.Type.GetType(ParseFieldType(pField.Type));
//字段默认值
pDataColumn.DefaultValue = pField.DefaultValue;
//当字段为String类型是设置字段长度
if (pField.VarType == 8)
{
pDataColumn.MaxLength = pField.Length;
}
//字段添加到表中
pDataTable.Columns.Add(pDataColumn);
pField = null;
pDataColumn = null;
}
return pDataTable;
}
```

因为GeoDatabase的数据类型与.NET的数据类型不同，故要进行转换。转换函数如下：

```
/// <summary>
/// 将GeoDatabase字段类型转换成.Net相应的数据类型
/// </summary>
/// <param name="fieldType">字段类型</param>
/// <returns></returns>
public static string ParseFieldType(esriFieldType fieldType)
{
switch (fieldType)
{
case esriFieldType.esriFieldTypeBlob:
return "System.String";
case esriFieldType.esriFieldTypeDate:
return "System.DateTime";
case esriFieldType.esriFieldTypeDouble:
```

```
return "System.Double";
case esriFieldType.esriFieldTypeGeometry:
return "System.String";
case esriFieldType.esriFieldTypeGlobalID:
return "System.String";
case esriFieldType.esriFieldTypeGUID:
return "System.String";
case esriFieldType.esriFieldTypeInteger:
return "System.Int32";
case esriFieldType.esriFieldTypeOID:
return "System.String";
case esriFieldType.esriFieldTypeRaster:
return "System.String";
case esriFieldType.esriFieldTypeSingle:
return "System.Single";
case esriFieldType.esriFieldTypeSmallInteger:
return "System.Int32";
case esriFieldType.esriFieldTypeString:
return "System.String";
default:
return "System.String";
}
}
```

1/3 1 2 3 下一页 尾页

10 来顶一下 返回首页

发表评论 共有 12条评论

用户名: 密码:

验证码: 3942 匿名发表

提交留言

网站首页 | 关于我们 | 服务条款 | 广告服务 | 联系我们 | 网站地图 | 免责声明 | 返回顶部

Copyright ? 2008-2009 3sdn.net, All rights reserved.

3sdnnet@Gmail.com



用户名: 密码:
登陆 注册
加入收藏 | 设为首页 | 会员中心 | 我要投稿 | RSS



《ArcGIS Engine+C#实例开发教程》第八讲 属性数据表的查询显示

时间: 2009-05-01 06:31:26 来源: 3SDN.Net 作者: 3SDN原创 点击量: 5517

3.装载DataTable数据

从上一步得到的DataTable还没有数据，只有字段信息。因此，我们要通过ICursor从ITable中逐一取出每一行数据，即IRow。再创建DataTable中相应的DataRow，根据IRow设置DataRow信息，再将所有的DataRow添加到DataTable中，就完成了DataTable数据的装载。

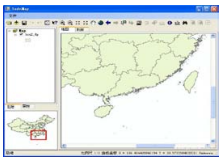
为保证效率，一次最多只装载2000条数据到DataGridView。函数代码如下:

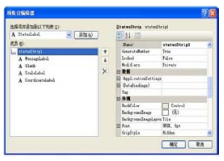
```
/// <summary>
/// 填充DataTable中的数据
/// </summary>
/// <param name="pLayer"></param>
/// <param name="tableName"></param>
/// <returns></returns>
public static DataTable CreateDataTable(ILayer pLayer, string tableName)
{
//创建空DataTable
DataTable pDataTable = CreateDataTableByLayer(pLayer, tableName);
//取得图层类型
string shapeType = getShapeType(pLayer);
//创建DataTable的行对象
DataRow pDataRow = null;
//从ILayer查询到ITable
ITable pTable = pLayer as ITable;
ICursor pCursor = pTable.Search(null, false);
//取得ITable中的行信息
IRow pRow = pCursor.NextRow();
```

推荐资讯


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实

相关文章

- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》教程Bug及优化方案1
- 《ArcGIS Engine+C#实例开发教程》第六讲 右键菜单添加
- 《ArcGIS Engine+C#实例开发教程》第五讲 鹰眼的实现
- 《ArcGIS Engine+C#实例开发教程》第四讲 状态栏信息的
- 《ArcGIS Engine+C#实例开发教程》第三讲 MapControl与
- 《ArcGIS Engine+C#实例开发教程》第二讲 菜单的添加及
- 《ArcGIS Engine+C#实例开发教程》第一讲 桌面GIS应用
- AO开发客户化组件（DLL+ArcMap）入门实例

栏目更新

- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发

```
int n = 0;
while (pRow != null)
{
//新建DataTable的行对象
pDataRow = pDataTable.NewRow();
for (int i = 0; i < pRow.Fields.FieldCount; i++)
{
//如果字段类型为esriFieldTypeGeometry，则根据图层类型设置字段值
if (pRow.Fields.get_Field(i).Type == esriFieldType.esriFieldTypeGeometry)
{
pDataRow[i] = shapeType;
}
//当图层类型为Anotation时，要素类中会有esriFieldTypeBlob类型的数据，
//其存储的是标注内容，如此情况需将对应的字段值设置为Element
else if (pRow.Fields.get_Field(i).Type == esriFieldType.esriFieldTypeBlob)
{
pDataRow[i] = "Element";
}
else
{
pDataRow[i] = pRow.get_Value(i);
}
}
//添加DataRow到DataTable
pDataTable.Rows.Add(pDataRow);
pDataRow = null;
n++;
//为保证效率，一次只装载最多条记录
if (n == 2000)
{
pRow = null;
}
else
{
pRow = pCursor.NextRow();
}
}
return pDataTable;
```


- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- （连载三）VC、OpenGL、ArcG
- （连载二）VC、OpenGL、ArcG
- （连载一）VC、OpenGL、ArcG

栏目热门

- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- ArcGIS Engine中的数据访问
- （连载一）VC、OpenGL、ArcG
- ArcGIS Engine的license部署

```
}
```

上面的代码中涉及到一个获取图层类型的函数getShapeTape，此函数是通过ILayer判断图层类型的，代码如下：

```
/// <summary>
/// 获得图层的Shape类型
/// </summary>
/// <param name="pLayer">图层</param>
/// <returns></returns>
public static string getShapeType(ILayer pLayer)
{
IFeatureLayer pFeatLyr = (IFeatureLayer)pLayer;
switch (pFeatLyr.FeatureClass.ShapeType)
{
case esriGeometryType.esriGeometryPoint:
return "Point";
case esriGeometryType.esriGeometryPolyline:
return "Polyline";
case esriGeometryType.esriGeometryPolygon:
return "Polygon";
default:
return "";
}
}
```

4.绑定DataTable到DataGridView

通过以上步骤，我们已经得到了一个含有图层属性数据的DataTable。现定义一个AttributeTableFrm类的成员变量：

```
public DataTable attributeTable;
```

通过以下函数，我们很容易将其绑定到DataGridView控件中。

```
/// <summary>
/// 绑定DataTable到DataGridView
/// </summary>
/// <param name="player"></param>
public void CreateAttributeTable(ILayer player)
{
string tableName;
tableName = getValidFeatureClassName(player .Name );
```

```
attributeTable = CreateDataTable(player,tableName );
this.dataGridView1 .DataSource = attributeTable ;
this.Text = "属性表[" + tableName + "] " + "记录数："+attributeTable.Rows.Count .ToString();
}
```

因为DataTable的表名不允许含有“.”，因此我们用“_”替换。函数如下：

```
/// <summary>
/// 替换数据表名中的点
/// </summary>
/// <param name="FCname"></param>
/// <returns></returns>
public static string getValidFeatureClassName(string FCname)
{
int dot = FCname.IndexOf(".");
if (dot != -1)
{
return FCname.Replace(".", "_");
}
return FCname;
}
```

2/3 首页 上一页 1 2 3 下一页 尾页

10 来顶一下 返回首页

发表评论 共有 12条评论

用户名: 密码:
验证码: 5570 匿名发表

提交留言

站内搜索: 文章 搜索 高级搜索

网站首页 | 关于我们 | 服务条款 | 广告服务 | 联系我们 | 网站地图 | 免责声明 | 返回顶部

Copyright ? 2008-2009 3sdn.net, All rights reserved.

3sdnnet@Gmail.com

用户名: 密码:
登陆 注册
加入收藏 | 设为首页 | 会员中心 | 我要投稿 | RSS



《ArcGIS Engine+C#实例开发教程》第八讲 属性数据表的查询显示

时间: 2009-05-01 06:31:26 来源: 3SDN.Net 作者: 3SDN原创 点击量: 5519

5.调用属性表窗体

通过1-4步骤，我们封装了一个AttributeTableFrm类，此类能够由ILayer显示图层中的属性表数据。那怎么调用AttributeTableFrm呢？

前面已经提到，我们是在TOCControl选中图层的右键菜单中弹出属性表窗体的，因此我们需要添加一个菜单项到TOCControl中Layer的右键菜单。而在第六讲中，我们采用的是AE中的IToolbarMenu实现右键菜单的，故我们还需自定义一个Command，实现打开属性表的功能。

以ArcGIS的Base Command为模板新建项“OpenAttributeTable.cs”。

注意：新建Base Command模板时，会弹出一个对话框让我们选择模板适用对象，这时我们要选择MapControl、PageLayoutControl，即选择第二项或者倒数第二项。

添加如下引用：

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

添加成员变量：

```
private ILayer m_pLayer;
```

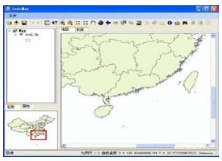
修改构造函数为：

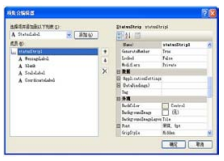
```
public OpenAttributeTable(ILayer pLayer)
{
//
// TODO: Define values for the public properties
//
```

推荐资讯


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实


《ArcGIS Engine+C#实

相关文章

- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》第七讲 图层符号选择
- 《ArcGIS Engine+C#实例开发教程》教程Bug及优化方案1
- 《ArcGIS Engine+C#实例开发教程》第六讲 右键菜单添加
- 《ArcGIS Engine+C#实例开发教程》第五讲 鹰眼的实现
- 《ArcGIS Engine+C#实例开发教程》第四讲 状态栏信息的
- 《ArcGIS Engine+C#实例开发教程》第三讲 MapControl与
- 《ArcGIS Engine+C#实例开发教程》第二讲 菜单的添加及
- 《ArcGIS Engine+C#实例开发教程》第一讲 桌面GIS应用
- AO开发客户化组件（DLL+ArcMap）入门实例

栏目更新

- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发
- 《ArcGIS Engine+C#实例开发

```
base.m_category = ""; //localizable text
base.m_caption = "打开属性表"; //localizable text
base.m_message = "打开属性表"; //localizable text
base.m_toolTip = "打开属性表"; //localizable text
base.m_name = "打开属性表"; //unique id, non-localizable (e.g. "MyCategory_MyCommand")
m_pLayer = pLayer;
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On_Click函数中添加如下代码，以创建并打开属性表窗体。

```
/// <summary>
/// Occurs when this command is clicked
/// </summary>
public override void OnClick()
{
// TODO: Add OpenAttributeTable.OnClick implementation
AttributeTableFrm attributeTable = new AttributeTableFrm();
attributeTable.CreateAttributeTable(m_pLayer);
attributeTable.ShowDialog();
}
```

至此，我们完成了OpenAttributeTable命令。显然，我们要在TOCControl的OnMouseDown事件中调用此命令。

因为，当前选中的图层参数，即ILayer是通过OpenAttributeTable的构造函数传入的，而选中的ILayer是动态变化的，所以我们无法在窗体初始化的Form1_Load事件中就添加OpenAttributeTable菜单项到右键菜单。但我们可以在OnMouseDown事件中动态添加OpenAttributeTable菜单项。

- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- （连载三）VC、OpenGL、ArcG
- （连载二）VC、OpenGL、ArcG
- （连载一）VC、OpenGL、ArcG

栏目热门

- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- 《**ArcGIS Engine+C#**实例开发
- ArcGIS Engine中的数据访问
- （连载一）VC、OpenGL、ArcG
- ArcGIS Engine的license部署

要注意的是，最后我们必须移除添加的OpenAttributeTable菜单项，不然每次按下右键都会添加此菜单项，将造成右键菜单中含有多个OpenAttributeTable菜单项。

修改TOCControl的OnMouseDown事件的部分代码如下：

```
 private void axTOCControl1_OnMouseDown(object sender, ITOCControlEvents_OnMouseDownEvent e)
{
//.....
//弹出右键菜单
if (item == esriTOCControlItem.esriTOCControlItemMap)
m_menuMap.PopupMenu(e.x, e.y, m_tocControl.hWnd);
if (item == esriTOCControlItem.esriTOCControlItemLayer)
{
//动态添加OpenAttributeTable菜单项
m_menuLayer.AddItem(new OpenAttributeTable(layer), -1, 2, true, esriCommandStyles.esriCommandStyleTextOnly);
m_menuLayer.PopupMenu(e.x, e.y, m_tocControl.hWnd);
//移除OpenAttributeTable菜单项，以防止重复添加
m_menuLayer.Remove(2);
}
}
```

6.编译运行

按下F5，编译运行程序，相信你已经实现了开篇处展示的属性表效果了吧！

以上代码在Windows XP Sp3 + VS2005 + AE9.2环境下编译通过。

下一讲中，我将给大家带来的是图层文字标注的实现，敬请关注3SDN.Net!

3/3 首页 上一页 1 2 3

10 来顶一下 返回首页

发表评论 共有 12条评论

用户名: 密码:

验证码: 0167 匿名发表

提交留言

站内搜索： 文章 搜索 高级搜索

网站首页 | 关于我们 | 服务条款 | 广告服务 | 联系我们 | 网站地图 | 免责声明 | 返回顶部

Copyright ? 2008-2009 3sdn.net, All rights reserved.

3sdnnet@Gmail.com

