第16章

空间数据库加载

16.1 通过设置属性加载个人数据库

首先通过 IPropertySet 接口定义要连接数据库的一些相关属性，在个人数据库中为数据库的路径，例如：

```
IPropertySet  Propset = new PropertySetClass();
Propset.SetProperty("DATABASE",@"D:\test\Ao\data\sh\MapData.mdb" );
```

定义完属性并设置属性后，就可以进行打开数据库的操作了。在 ArcEngine 开发中，存在 IWorkspaceFactory、IFeatureWorkspace、IFeatureClass、IFeatureLayer 等几个常用的用于打开和操作数据空间地物的接口。IWorkspaceFactory 是一个用于创建和打开工作空间的接口，它是一个抽象的接口，我们在具体应用时，要用对应的工作空间实例化它，方式如下：

```
IWorkspaceFactory Fact = new AccessWorkspaceFactoryClass ();
```

完成了工作空间的实例化后，就可以根据上面设置的属性打开对应的 Access 数据库。打开方式如下：

```
IFeatureWorkspace Workspace = Fact.Open(Propset,0) as IFeatureWorkspace;
IFeatureClass Fcls = Workspace.OpenFeatureClass("District");
IFeatureLayer Fly = new FeatureLayerClass();
Fly.FeatureClass = Fcls;
MapCtr.Map.AddLayer (Fly);
MapCtr.ActiveView.Refresh();
```

其中 District 为地物类的名字，MapCtr 为 AE 中 MapControl 的对象。上面的通过属性设置加载数据空间的方式还可用于 SDE 数据库，在 SDE 数据库加载时会介绍。

以下为通过设置属性加载 Access 数据库的完整 C#代码。

```
public void AddAccessDBByPro()
```

```
{
    IPropertySet  Propset = new PropertySetClass();
    Propset.SetProperty("DATABASE",@"D:\test\Ao\data\sh\MapData.mdb" );
    IWorkspaceFactory Fact = new AccessWorkspaceFactoryClass ();
    IFeatureWorkspace Workspace = Fact.Open(Propset,0) as IFeatureWorkspace;
    IFeatureClass Fcls = Workspace.OpenFeatureClass ("District");
    IFcatureLayer Fly = new FeatureLayerClass();
    Fly.FeatureClass = Fcls;
    MapCtr.Map.AddLayer(Fly);
    MapCtr.ActiveView.Refresh();
}
```

16.2 通过名称加载个人数据库

通过名称加载个人数据，通常使用 IWorkspaceName 接口提供的方法和属性，这里先给出完整的代码。

```
public void AddAccessDBByName()
{
    IWorkspaceName  pWorkspaceName = new WorkspaceNameClass() ;
                    pWorkspaceName.WorkspaceFactoryProgID= "esriDataSourcesGDB.
AccessWorkspace Factory";
    pWorkspaceName.PathName = @"D:\test\Ao\data\sh\MapData.mdb";
    IName n = pWorkspaceName as IName ;
    IFeatureWorkspace Workspace = n.Open() as IFeatureWorkspace;
    IFeatureClass Fcls = Workspace.OpenFeatureClass ("District");
    IFeatureLayer Fly = new FeatureLayerClass();
    Fly.FeatureClass = Fcls;
    MapCtr.Map.AddLayer (Fly);
    MapCtr.ActiveView.Refresh();
}
```

打开 Access 工作空间后，接下来的代码是一样的，都是找到对应的地物类，赋给相应的层，通过 MapControl 控件添加对应的层，然后刷新地图。现在讲解一下上面的代码，首先是创建一个个人数据库工作空间名，再指定工作空间名的 ProgID，以确定打开的是什么类型的工作空间。例如在打开 Access 个人数据库时，使用的是下面的代码：

```
  IworkspaceName pWorkspaceName = new WorkspaceNameClass() ;
   pWorkspaceName.WorkspaceFactoryProgID= "esriDataSourcesGDB.AccessWorkspaceFactory";
pWorkspaceName.PathName = @"D:\test\Ao\data\sh\MapData.mdb";
```

属性 WorkspaceFactoryProgID 可以确保工作空间是 AccessWorkspaceFactory，即个人数据库，同时指定要打开数据库的路径。为了打开数据库，通过 AE 的类图可以看到，打开工作空间必须使用 IName 接口，所以接着定义 IName 对象，把工作空间名转换成 IName 类型，并赋值给 IName 对象，然后通过 IName 对象的 Open()方法打开相应的工作空间，代码如下：

```
IName n = pWorkspaceName as IName ;
IFeatureWorkspace Workspace = n.Open () as IFeatureWorkspace;
```

16.3 SDE 数据库

ESRI 公司为了使空间数据能保存在关系数据库中，并且能很好地查询相关的空间属性，而开发了一个中间件，使用 SDE 能很好地将空间数据保存在关系数据库中，如 Orcale、SQL Server 等。要了解 SDE 具体的细节，可查找相关的资料，这里只介绍一下 SDE 数据库连接加载。SDE 数据库的联机分为直接连接和通过 SDE 连接。当服务器的性能比较好的时候，可以采用 SDE 连接，否则采用直接连接，这样可以减轻服务器的工作。建议采用直接连接。其实，SDE 连接方式和直接连接的方式只是一个属性参数设置的问题。与个人数据库采用属性连接的方式一样，可先定义一个属性对象，设置属性参数，接着定义一个工作空间，并用 SdeWorkspaceFactoryClass() 实例化它，然后加在加载图层即可。至于加载图层的代码，与加载个人数据库中图层的方法一样。其实不止加载这两种数据类型，加载其他类型的数据时也是采用相同的方法加载图层，只是工作空间采用不同的实例而已。

```
public void AddSDELayer(bool ChkSdeLinkModle)
{
    //定义一个属性
    IPropertySet  Propset = new PropertySetClass();
    if (ChkSdeLinkModle==true)     // 采用 SDE 连接
    {
        //设置数据库服务器名，服务器所在的 IP 地址
        Propset.SetProperty ("SERVER", "192.168.2.41");
        //设置 SDE 的端口，这是安装时指定的，默认安装时"port:5151"
        Propset.SetProperty ("INSTANCE", "port:5151");
        //SDE 的用户名
        Propset.SetProperty ("USER", "sa");
        //密码
        Propset.SetProperty ("PASSWORD", "sa");
        //设置数据库的名字，只有 SQL Server  Informix 数据库才需要设置
        Propset.SetProperty ("DATABASE", "sde");
        //SDE 的版本，在这里为默认版本
        Propset.SetProperty ("VERSION", "SDE.DEFAULT");
    }
    else      // 直接连接
    {
        //设置数据库服务器名，如果是本机，可以用"sde:sqlserver:."
        Propset.SetProperty ("INSTANCE", "sde:sqlserver:zhpzh");
        //SDE 的用户名
        Propset.SetProperty ("USER", "sa");
        //密码
        Propset.SetProperty ("PASSWORD", "sa");
        //设置数据库的名字，只有 SQL Server  Informix 数据库才需要设置
        Propset.SetProperty ("DATABASE", "sde");
        //SDE 的版本，在这里为默认版本
        Propset.SetProperty ("VERSION", "SDE.DEFAULT");
    }
    //定义一个工作空间，并实例化为 SDE 的工作空间
    IWorkspaceFactory Fact = new SdeWorkspaceFactoryClass();
    //打开 SDE 工作空间，并转化为地物工作空间
```

```
        IFeatureWorkspace Workspace = (IFeatureWorkspace )Fact.Open(Propset,0);
        /*定义一个地物类，并打开 SDE 中的管点地物类，写的时候一定要写全，如 SDE 中有一个管点层，就不能写成 IFeatureClass Fcls = Workspace.OpenFeatureClass ("管点"); 这样，而一定要写成下面的样子*/
          IFeatureClass Fcls = Workspace.OpenFeatureClass ("sde.dbo.管点");
          IFeatureLayer Fly = new FeatureLayerClass ();
          Fly.FeatureClass = Fcls;
          MapCtr.Map.AddLayer (Fly);
          MapCtr.ActiveView.Refresh ();
      }
```

16.4 分图层加载 CAD 图层

我们可以把 CAD 图分为点、线、面、标注等加载到 MapControl 中。与加载其他数据一样，首先要定义一个工作空间，并用 CadWorkspaceFactoryClass ( ) 实例化它，当得到了工作空间后，就可以打开相应的工作空间，然后再打开指定的层类型。完整的代码如下：

```
public void AddCADByLayer()
 {
   //定义工作空间，并用 CadWorkspaceFactoryClass()实例化它
    IWorkspaceFactory Fact = new CadWorkspaceFactoryClass();
    //打开相应的工作空间，并赋值给要素空间，OpenFromFile ( )
    //中的参数为 CAD 文件夹的路径
IFeatureWorkspace Workspace = Fact.OpenFromFile(@"I:\test\",0)  as IFeatureWorkspace;
 /*打开线要素类，如果要打开点类型的要素，需要把下面的代码改成：
    IFeatureClass Fcls = Workspace.OpenFeatureClass ("modle.dwg:point");
```

由此可见，modle.dwg 为 CAD 图的名字，后面加上要打开的要素类的类型，中间用冒号隔开。对于多义线，可以用以下代码打开：

```
    IFeatureClass Fcls = Workspace.OpenFeatureClass ("modle.dwg:polyline");
     IFeatureLayer Fly = new FeatureLayerClass ();
     Fly.FeatureClass = Fcls;
     MapCtr.Map.AddLayer (Fly);
     MapCtr.ActiveView.Refresh ();
}
```

16.5 整幅 CAD 图的加载

当要加载整幅 CAD 图时，需要使用下面的代码，这和加载地物类有一定的区别，详细的介绍请看代码中的注释：

```
public void AddWholeCAD()
{
    //下面的两行代码是定义一个 CAD 工作空间，然后打开它，但这次不是赋值给
    //IFeatureWorkspace 对象，而是赋值给 IWorkspace 定义的对象
    IWorkspaceFactory Fact = new CadWorkspaceFactoryClass();
    IWorkspace Workspace = Fact.OpenFromFile(@"I:\test\",0);
    //定义一个 CAD 画图空间，并把上面打开的工作空间赋给它
    ICadDrawingWorkspace dw =  Workspace as ICadDrawingWorkspace;
```

```
    //定义一个 CAD 的画图数据集，并且打开上面指定的工作空间中的一幅 CAD 图，
    //然后赋值给 CAD 数据集
    ICadDrawingDataset ds = dw.OpenCadDrawingDataset ("modle.DWG");
    //通过 ICadLayer 类，把上面得到的 CAD 数据赋值给 ICadLayer 类对象的
    //CadDrawingDataset 属性
    ICadLayer CadLayer = new  CadLayerClass();
    CadLayer.CadDrawingDataset = ds;
    //利用 MapControl 加载 CAD 层
    MapCtr.Map.AddLayer (CadLayer);
    MapCtr.ActiveView.Refresh ();
}
```

上面的代码通过将 CAD 数据 DWG 整个添加到地图控件中，加载时 ArcEngine 就会将 DWG 数据文件中同一类型的数据作为一个图层添加到地图控件中。

为了在 MapControl 中加载 CAD 层，必须使用 ICadLayer 控件的对象，因为在 MapCtr.Map.AddLayer ()方法中只能是 ICadLayer 的对象。