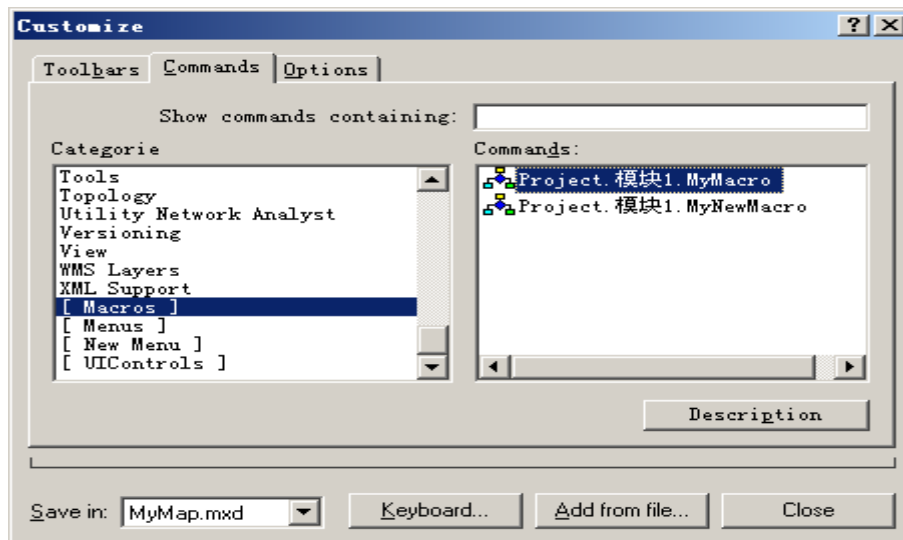
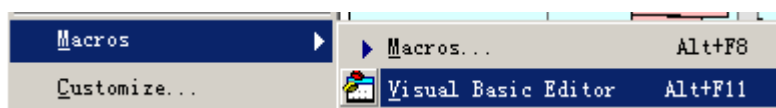


1. ArcMap 的定制与开发

- A、单击 Tools—>Customizes... 命令，打开定制对话框后，可以对 ARCMAP 中的工具条，菜单项，进行定制，可以加载用户自定义的宏、VBA 程序、DLL 命令等。

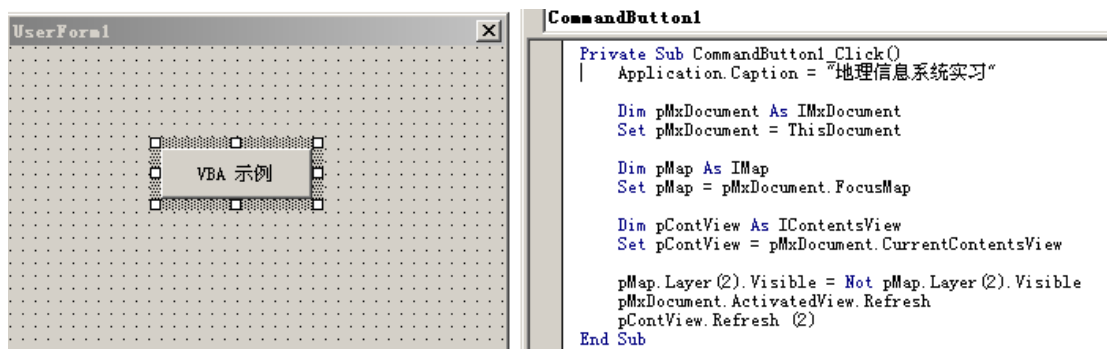


- B、定制 VBA 程序。Tools—>Visual Basic Editor... 打开编辑器：

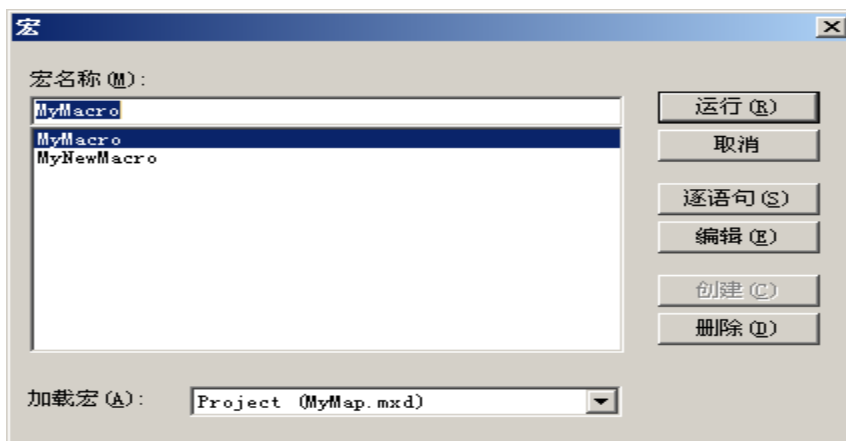


在 Project (NewMap.mxd) 下插入一个新的 UserForm1。并在新建的窗体中添加一个命令按钮，Caption 为 VB 窗体示例。

双击命令按钮为其添加点击事件程序。

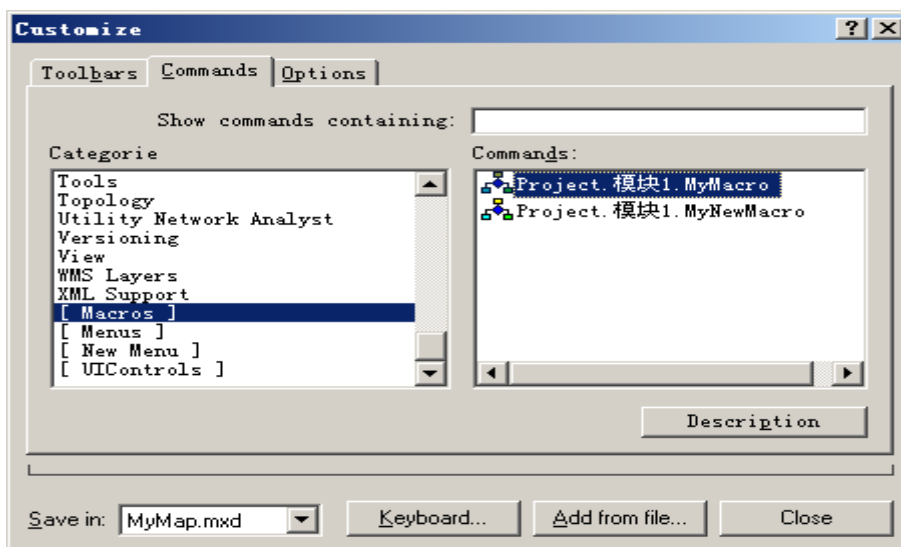
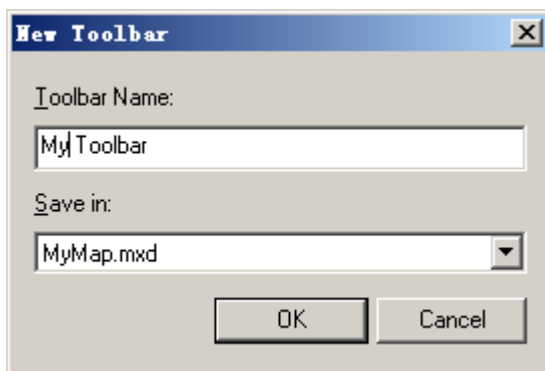


- C、Tools—>Macros... 打开宏对话框。并新建一个名为 MyMacro 的宏，其中加载宏 (A) 中选择 Project(NewMap.mxd)。并在 VBA 编辑器中为该宏编写代码。其中宏 MyMacro1 直接运行上面的 VBA 窗体



D、打开 Customize 对话框，选择 Commands 栏。通过 New Uicontrol，用户可以定制 4 类命令，按钮、编辑框、工具命令、组合框。

在此先创建一个 NewToolbar：



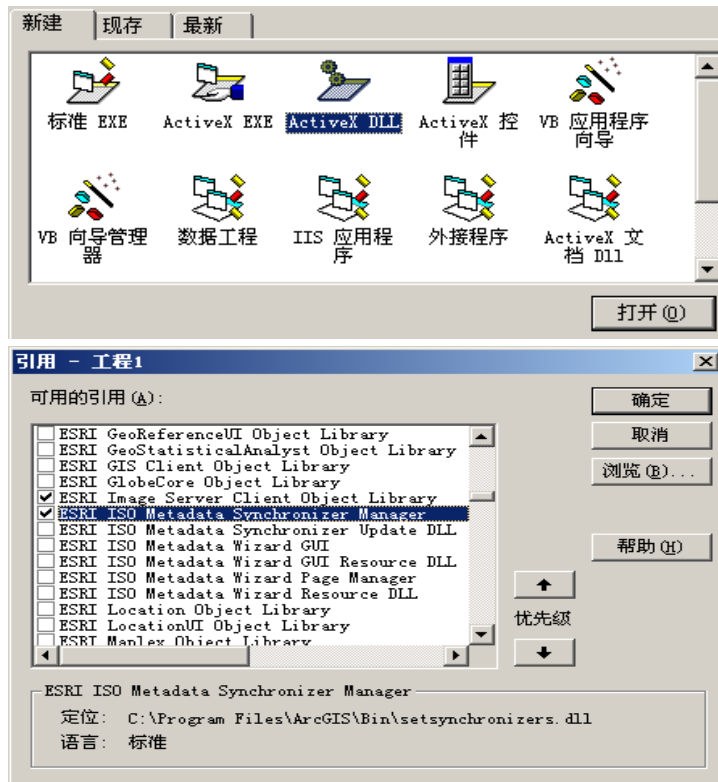
将刚才的宏拖到新建的工具条即可。

E、编写 VB 的 DLL 动态链接库。

上面介绍的定制方法必须在 ARCMAP 环境下进行，离不开环境，且代码随文档保存，难以进行深层次开发，代码解释执行，效率差，保密性差，但相对容易些，ARCMAP 已经提供了现成的对象与接口。下面开始介绍通过 DLL 操作 ArcObjects 定制 ARCGIS 命令。这种开发模式，最好通过改写已做好的示例程序来实现，不必要自己编写所有的代码。建议用 VB 编写 DLL，其他语言也可，但范例代码少。

打开 VB，新建 ActiveX DLL 工程。

在工程菜单下打开引用对话框，为该工程添加 ESRI 的库文件。其中 ESRI Object library 项一定要引用。



工程中加入一 VB 窗体 frmresources，放置一图片框，图片框内容为一个图标。

加入一类模块，该模块包含了 ARCOBJECTS 接口 ICOMMAND。接口是 ARCGIS 定制与开发的关键，通过引用不同的接口，类可以实现不同的功能，不同的功能组成了 ARCGIS 的程序框架。类的实现代码如下：

Option Explicit

Implements ICommand

Private m_pApp As IApplication

Private Property Get ICommand_BitMap() As esriSystem.OLE_HANDLE

ICommand_BitMap = frmResources.Picture1.Picture

End Property

Private Property Get iCommand_Caption() As String

iCommand_Caption = "VB Souldtion"

End Property

Private Property Get iCommand_Category() As String

iCommand_Category = "MyVBSolution"

End Property

```
Private Property Get ICommand_Checked() As Boolean
    ICommand_Checked = False
End Property
```

```
Private Property Get ICommand_Enabled() As Boolean
    ICommand_Enabled = True
End Property
```

```
Private Property Get ICommand_HelpContextID() As Long
    '该命令的帮助索引
End Property
```

```
Private Property Get ICommand_HelpFile() As String
    '该命令的帮助文档
End Property
```

```
Private Property Get ICommand_Message() As String
    ICommand_Message = "这是地信实习程序示例"
End Property
```

```
Private Property Get ICommand_Name() As String
    ICommand_Name = "test"
End Property
```

```
Private Sub ICommand_OnClick()
    m_pApp.Caption = "地理信息系统实习"

    Dim pMxDocument As IMxDocument
    Set pMxDocument = m_pApp.Document

    Dim pMap As IMap
    Set pMap = pMxDocument.FocusMap

    Dim pContView As IContentView
    Set pContView = pMxDocument.CurrentContentView

    pMap.Layer(2).Visible = Not pMap.Layer(2).Visible
    pMxDocument.ActiveView.Refresh
    pContView.Refresh (2)
End Sub
```

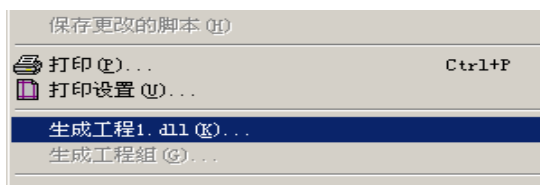
```
Private Sub ICommand_OnCreate(ByVal hook As Object)
    Set m_pApp = hook
End Sub
```

Private Property Get ICommand_Tooltip() As String

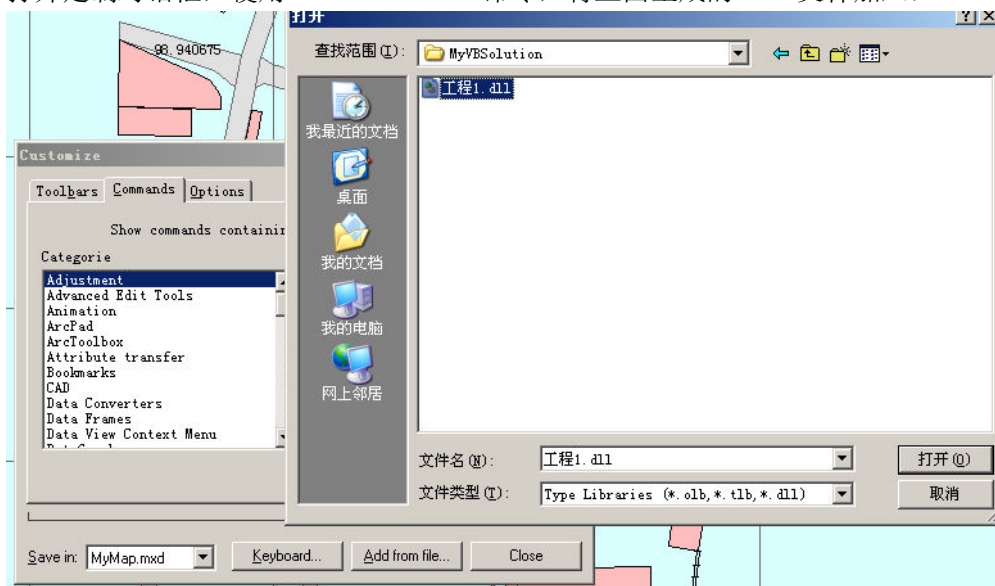
ICommand_Tooltip = "地信实习，ArcObjects 示例程序"

End Property

编译生成 DLL 文件，注意在有多个生成版本时，工程选项中最好选择该 DLL 为二进制兼容，这样保证每次生成的 DLL 具有同一注册值，这样，可以多次编译，但在 ARCGIS 中只需加载一次。不必重复加载：



打开定制对话框，使用 add from file 命令，将上面生成的 DLL 文件加入：



工具条中的圆按钮是刚才生成的 UIButtonControl 命令，现在一起把它加到我的工具条上在上面自己创建的工具条上也可以添加 ArcMap 的标准菜单项。例如平移、前一视图等。



F、VB 应用程序中如何控制 ArcGIS 的客户端组件。

打开 VB 新建一 EXE 工程，在工程一>引用中添加 ESRI 的类库。创建一个窗体，再在窗体上添加一个命令按钮，Caption 属性为“VB 操作 ArcMap 示例”。



```

Dim m_pApp As Application
Dim pMxDocument As IMxDocument
Private Sub Command1_Click()
    Dim pMap As IMap
    Set pMap = pMxDocument.FocusMap

    Dim pContView As IContentsView
    Set pContView = pMxDocument.CurrentContentsView

    pMap.Layer(2).Visible = Not pMap.Layer(2).Visible
    pMxDocument.ActiveView.Refresh
    pContView.Refresh(2)
End Sub

Private Sub Form_Load()
    Dim m_appROT As IAppROT
    Set m_appROT = New AppROT
    Dim i As Integer
    If m_appROT.Count > 0 Then
        ' 并获取各程序变量
        For i = 0 To (m_appROT.Count - 1)
            If TypeOf m_appROT.Item(i) Is IGxApplication Then
                MsgBox "ArcCatalog is running"
            ElseIf TypeOf m_appROT.Item(i) Is IMxApplication Then
                Set m_pApp = m_appROT.Item(i)
                m_pApp.Caption = "地理信息系统实习"
                m_pApp.OpenDocument "D:\计算机地图制图实习\newTask\MyMap.mxd"
                Set pMxDocument = m_pApp.Document
                m_pApp.Visible = True

                Exit For
            End If
        Next i
    Else
        Set pMxDocument = New MxDocument

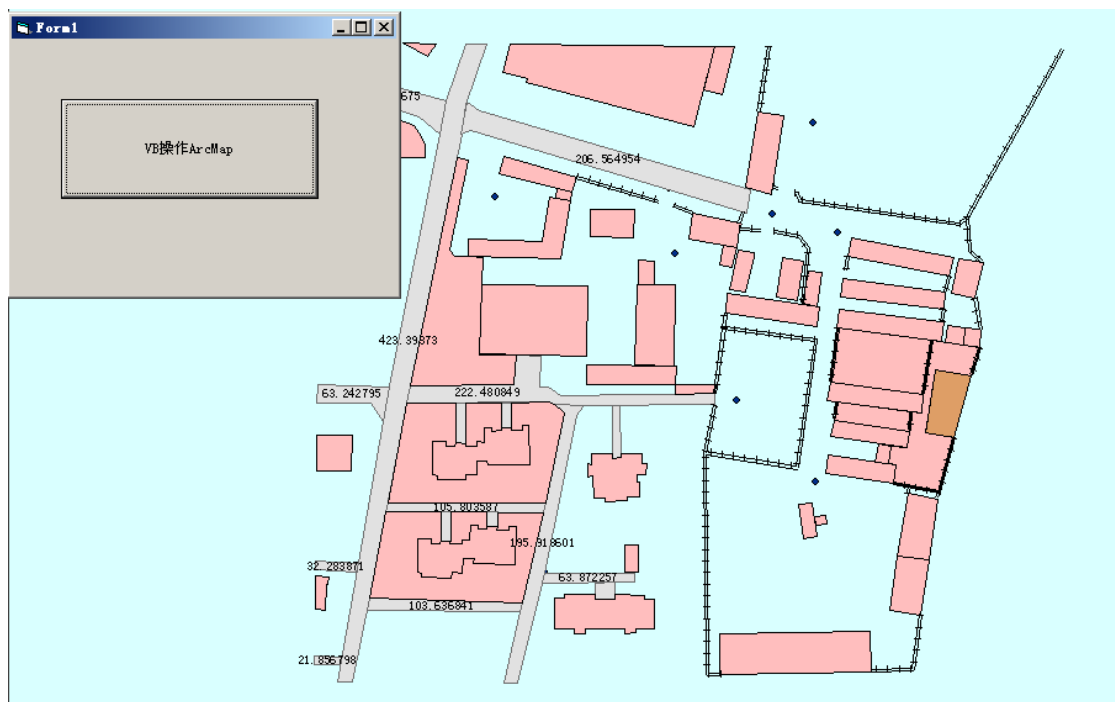
        Dim pMxDoc As MxDocument
        Set pMxDoc = pMxDocument
        Set m_pApp = pMxDoc.Parent
        m_pApp.Visible = True

        m_pApp.Caption = "地理信息系统实习"
        m_pApp.OpenDocument "D:\计算机地图制图实习\newTask\MyMap.mxd"
        Set pMxDocument = m_pApp.Document
    End If
End Sub

```

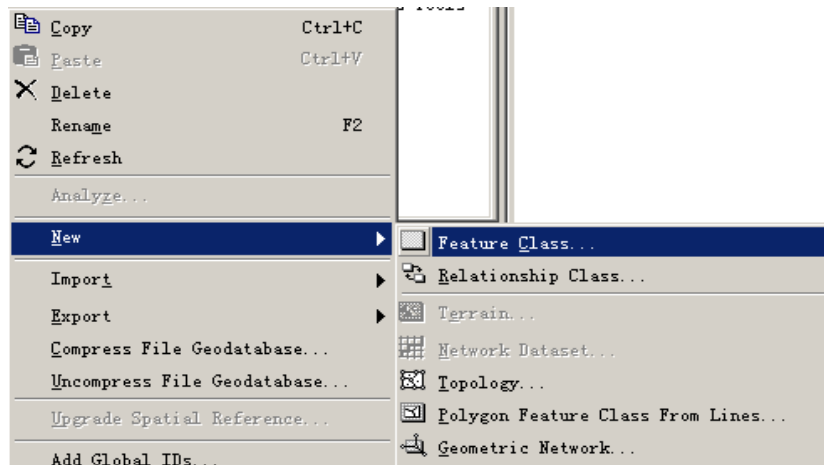
以上代码实现的功能是：

当应用程序开始运行进行窗体的加载时，对当前是否有 ArcGIS 客户端正在运行进行检查，如果检查到 ArcCatalog 或是 ArcScene 在运行则分别提示“ArcCatalog/ArcScene is running”；如果检查到 ArcMap 正在运行，则用该进程打开指定的文档；否则运行 ArcMap 新实例打开指定文档。运行界面如下：



G、开发自动添加外业测量数据的程序。

下面开发的程序具有添加外业数据的功能,以测量控制点的添加为例。添加外部数据时,可以采用人工操作添加,也可以采用程序自动添加,一般,对于大批量数据,手工添加速度较慢,当 ArcGIS 提供的功能不能完全满足要求时,可以考虑程序实现。步骤如下: 先创建容器,用来存放外业数据点,就是在 GEODATABASE 中新建一个要素类,命名为离散点:




```

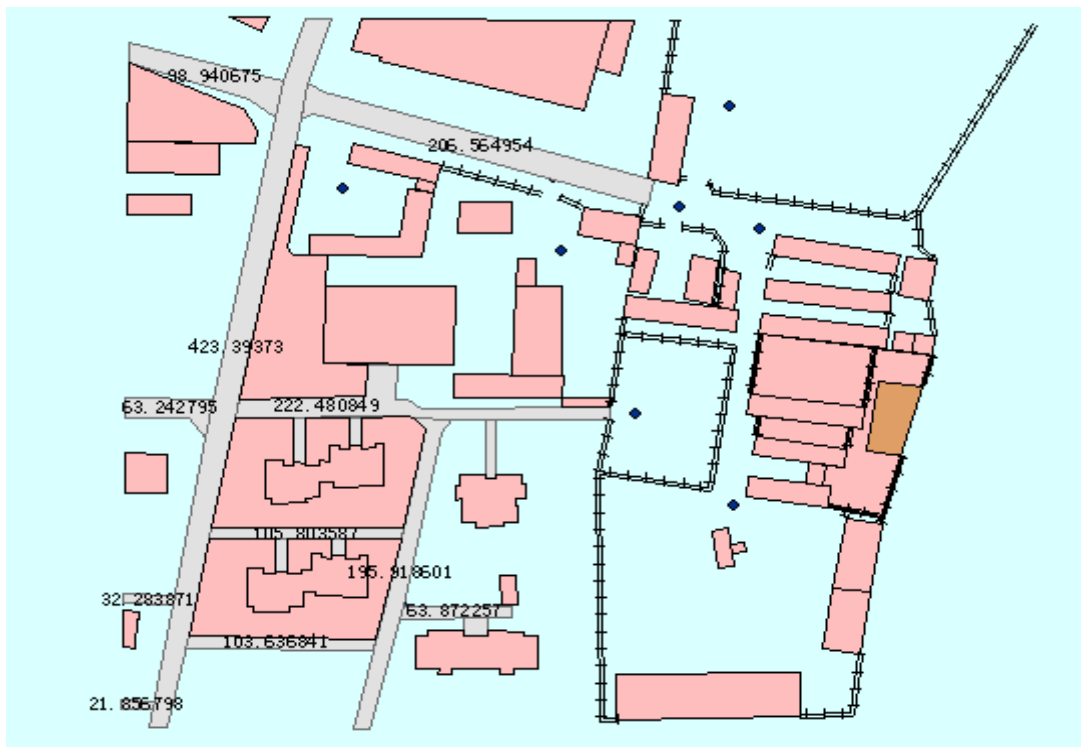
Private Sub UIButtonControl2_Click()
    Dim pDataset As IDataset
    Dim pFeatcls As IFeatureClass
    Dim pFeatLayer As IFeatureLayer
    Dim pDoc As IMxDocument
    Dim pMap As IMap
    Set pDoc = ThisDocument
    Set pMap = pDoc.FocusMap
    Set pFeatLayer = pMap.Layer(0) '当有多层数据时，要搜索所有数据层，根据名称找到要添加的数据层
    Set pFeatcls = pFeatLayer.FeatureClass
    Set pDataset = pFeatcls
    Dim pEditor As IEditor '引用编辑对象，对要素进行编辑
    Dim pID As New UID
    pID = "esriCore.Editor"
    Set pEditor = Application.FindExtensionByCLSID(pID)
    If pEditor.EditState = esriStateEditing Then Exit Sub

    pEditor.StartEditing pDataset.Workspace '开始编辑操作，相当于在editor 工具条中按start editing
    pEditor.StartOperation
    Dim pFeature As IFeature
    '打开数据文件，读取数据；
    Dim FileObj As FileSystemObject
    Dim TxtFile As TextStream
    Set FileObj = New FileSystemObject
    Set TxtFile = FileObj.OpenTextFile("D:\计算机地图制图实习\newTask\GCPs.txt", ForReading, False, TristateFalse)
    Dim strline As String
    Dim pX, pY, pZ As Double
    Dim pPoint As IPoint '定义点接口
    Dim pZAware As IZAware '定义点Z 值接口
    Do '循环读取文件数据
        Set pPoint = New Point
        Set pZAware = pPoint
        pZAware.ZAware = True
        strline = TxtFile.ReadLine
        pX = CDBl(Mid(strline, 1, 11)) '具体读取方式与文件存储格式相关，需要编程实现
        pY = CDBl(Mid(strline, 13, 10))
        pZ = CDBl(Mid(strline, 24, 5))
        pPoint.X = pX: pPoint.Y = pY: pPoint.Z = pZ
        Set pFeature = pFeatcls.CreateFeature
        Set pFeature.Shape = pPoint
        pFeature.Store '保存pfeature
    Loop Until (TxtFile.AtEndOfStream)
    TxtFile.Close '关闭文件
    pEditor.StopEditing True
    pDoc.ActivatedView.Refresh
End Sub

```

由于原始的 txt 数据文件点太多，而且与我定义的坐标不匹配，所以我修改了一下 txt 并减少了点的数量。添加点前后效果如下所示：

命令执行前：



执行添加点 LSD 命令后：

