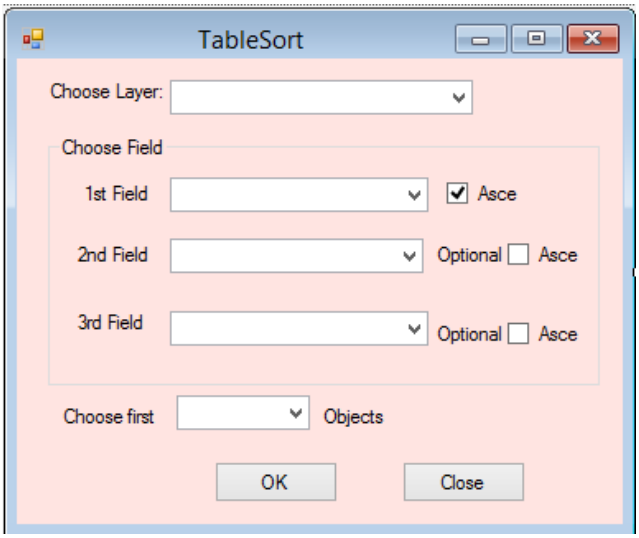
Chapter 11 TableSort

1. Add a Windows Form “TableSort” to the project:



GUI implementation detail could be found at “TableSort.Designer.cs” file.

2. Implementation of TableSort:

- Class member and Construction method:

```
IMapControl3 m_mapcontrol;
IMap m_map;
public TableSort(IHookHelper hook)
{
    if (hook != null)
    {
        m_mapcontrol = hook.Hook as IMapControl3;
        m_map = m_mapcontrol.Map;
        InitializeComponent();
    }
}
```

- Add GetLayers, GetFeatureLayer, getFeatureCount, CbxFieldAdditems, CbxLayersAddItems:

```
private IEnumLayer GetLayers()
{
    UID uid = new UIDClass();
    uid.Value = "{40A9E885-5533-11d0-98BE-00805F7CED21}";
    if (m_map.LayerCount != 0)
    {
        IEnumLayer layers = m_map.get_Layers(uid, true);
        return layers;
    }
    return null;
}
```

```
private void CbxLayersAddItems()
{
    if (GetLayers() == null) return;
    cbxLayers.Items.Clear();
    IEnumLayer layers = GetLayers();
    layers.Reset();
    ILayer layer = layers.Next();
    while (layer != null)
    {
        if (layer is IFeatureLayer)
        {
            cbxLayers.Items.Add(layer.Name);
        }
        layer = layers.Next();
    }
    cbxLayers.SelectedIndex = 0;
}

private IFeatureLayer GetFeatureLayer(string layerName)
{
    if (GetLayers() == null) return null;
    IEnumLayer layers = GetLayers();
    layers.Reset();
    ILayer layer = null;
    while ((layer = layers.Next()) != null)
    {
        if (layer.Name == layerName)
            return layer as IFeatureLayer;
    }
    return null;
}

private void getFeatureCount(string layername)
{
    int count = 0;
    IFeatureLayer featurelayer = GetFeatureLayer(layername);
    if (featurelayer == null) return;
    count = featurelayer.FeatureClass.FeatureCount(null);
    comboBox1.Items.Clear();
    for (int i = 1; i <= count; i++) comboBox1.Items.Add(i.ToString());
}

private void CbxFieldAdditems(IFeatureLayer featureLayer)
{
    IFields fields = featureLayer.FeatureClass.Fields;
    cbxFirstField.Items.Clear();
    cbxSecField.Items.Clear();
    cbxThirdField.Items.Clear();
    for (int i = 0; i < fields.FieldCount; i++)
    {
        cbxFirstField.Items.Add(fields.get_Field(i).Name);
        cbxSecField.Items.Add(fields.get_Field(i).Name);
        cbxThirdField.Items.Add(fields.get_Field(i).Name);
    }
    cbxFirstField.SelectedIndex = 0;
}
```

- Add selectedIndexChanged event of “cbxLayers” and Load event of form:

```
string layername;
IFeatureLayer featurelayer;
private void cbxLayers_SelectedIndexChanged(object sender, EventArgs e)
{
    layername = cbxLayers.Text;
    featurelayer = GetFeatureLayer(layername);
    CbxFieldAdditems(featurelayer);
    getFeatureCount(layername);
}

private void TableSort_Load(object sender, EventArgs e)
{
    CbxLayersAddItems();
    if (featurelayer == null) return;
    CbxFieldAdditems(featurelayer);
    getFeatureCount(layername);
}
```

- FeatureSort:

```
private void FeatureSort()
{
    IFeatureSelection featureSel = (IFeatureSelection)featurelayer;
    featureSel.SelectFeatures(null, esriSelectionResultEnum.
        esriSelectionResultNew, false);
    ISelectionSet selectionSet = featureSel.SelectionSet;
    ITableSort pTableSort = new ESRI.ArcGIS.Geodatabase.TableSort()
        as ITableSort;
    if(cbxSecField.Text==""&&cbxThirdField.Text=="")
    {
        pTableSort.Fields=cbxFirstField.Text;
        if(checkBox1.Checked==true)
            pTableSort.set_Ascending(cbxFirstField.Text,true);
    }
```

```
        if (cbxFirstField.Text != "" && cbxThirdField.Text != "")
        {
            pTableSort.Fields = cbxThirdField.Text + "," +
                cbxSecField.Text + "," + cbxThirdField.Text;
            if (checkBox1.Checked == true)
                pTableSort.set_Ascending(cbxFirstField.Text, true);
            else
                pTableSort.set_Ascending(cbxFirstField.Text, false);
            if (checkBox2.Checked == true)
                pTableSort.set_Ascending(cbxSecField.Text, true);
            else
                pTableSort.set_Ascending(cbxSecField.Text, false);
            if (checkBox3.Checked == true)
                pTableSort.set_Ascending(cbxSecField.Text, true);
            else
                pTableSort.set_Ascending(cbxThirdField.Text, false);
        }
        if (cbxSecField.Text != "" && cbxThirdField.Text == "")
        {
            pTableSort.Fields = cbxFirstField.Text + ","
                + cbxSecField.Text;

            if (checkBox1.Checked == true)
                pTableSort.set_Ascending(cbxFirstField.Text, true);
            else
                pTableSort.set_Ascending(cbxFirstField.Text, false);
            if (checkBox2.Checked == true)
                pTableSort.set_Ascending(cbxSecField.Text, true);
            else
                pTableSort.set_Ascending(cbxSecField.Text, false);
        }
        pTableSort.SelectionSet = selectionSet;
        pTableSort.Sort(null);
        IFeatureCursor featureCursor = (IFeatureCursor)pTableSort.Rows;
        IFeature feature = featureCursor.NextFeature();
        m_map.ClearSelection();
        int i = Convert.ToInt32(comboBox1.Text); int j = 0;
        while (feature != null)
        {
            j++;

           if (j <= i)
           {
               m_map.SelectFeature(featurelayer as ILayer, feature);
               feature = featureCursor.NextFeature();
           }
           else break;
       }
       m_mapcontrol.ActiveView.PartialRefresh(esriViewDrawPhase.
           esriViewGeoSelection, null, null);
}
```

- Click event of “OK” and “Cancel”:

```
private void btnOK_Click(object sender, EventArgs e)
{
    if (comboBox1.Text == "") return;
    FeatureSort();
}

private void button1_Click(object sender, EventArgs e)
{
    this.Dispose();
}
```

3. Add A Base Command Class TableSortCmd, and implement OnClick method:

```
public override void OnClick()
{
    // TODO: Add TableSortCmd.OnClick implementation
    TableSort form = new TableSort(m_hookHelper);
    form.Show(m_hookHelper as System.Windows.Forms.IWin32Window);
}
```

4. Back to MainForm, add a menu content and its click event:

```
private void tableSortToolStripMenuItem_Click(object sender, EventArgs e)
{
    ICommand command = new TableSortCmd();
    command.OnCreate(m_mapControl.Object);
    command.OnClick();
}
```