Shapefile label operations Tutorial (7)

**Tutorial (7)**

Purpose of this tutorial:

1. Create a label to the shape file from its attribute table column. User can select any attribute from shape file's attribute table.

2. Add a custom label to the shape files via user input. User can type any name in the text box and select the shape to set the label.

3. Change the label appearance.

**Step 1:** Add the DotSpatial reference and change the compile option.

Add the following references:

DotSpatial.Data.Forms.dll, DotSpatial.Symbology.dll, DotSpatial.Controls.dll, DotSpatial.Projections.dll, DotSpatial.Data.dll, DotSpatial.Topology.dll

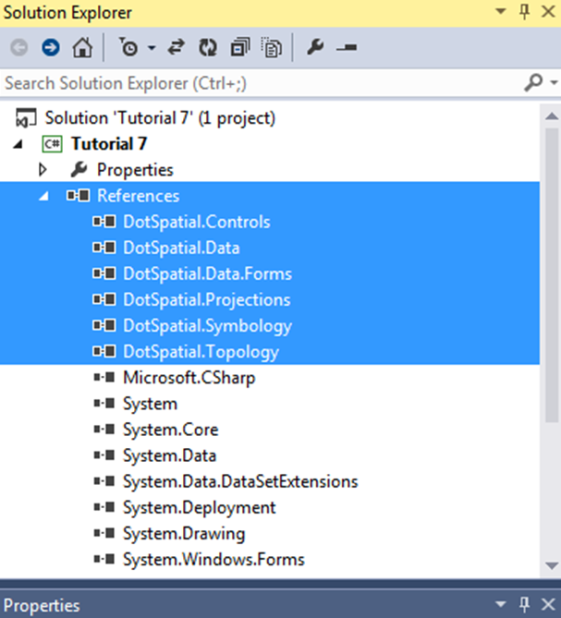


Figure 1: Required dlls

**Step 2:** Add the DotSpatial controls on the Visual Studio Toolbox window.

If you don't have DotSpatial control tab, create a new tab on the toolbox and add the DotSpatial controls to it, the same as the first tutorial.

**Step 3**: Design the Graphical user interface

Design the GUI as follows:

Add a menu strip control and design the menu items as follows:

**Main menu items**:

File, Map Options.

**Sub menu items:**

File menu has the following sub menu items:

Load, Clear and Exit

Map Options menu item has the following sub menu items:

Zoom In, Zoom Out, Zoom to MaxExtent, Pan, and Default.

Add a SpatialDockManager and set the properties as follows:

Name: sdmOperationsMap, Dock: Fill, BorderStyle: FixedSingle

Add a FontDialog box and ColorDialog box. Both dialog boxes are located on the tool box under the AllWindowsForms tab.

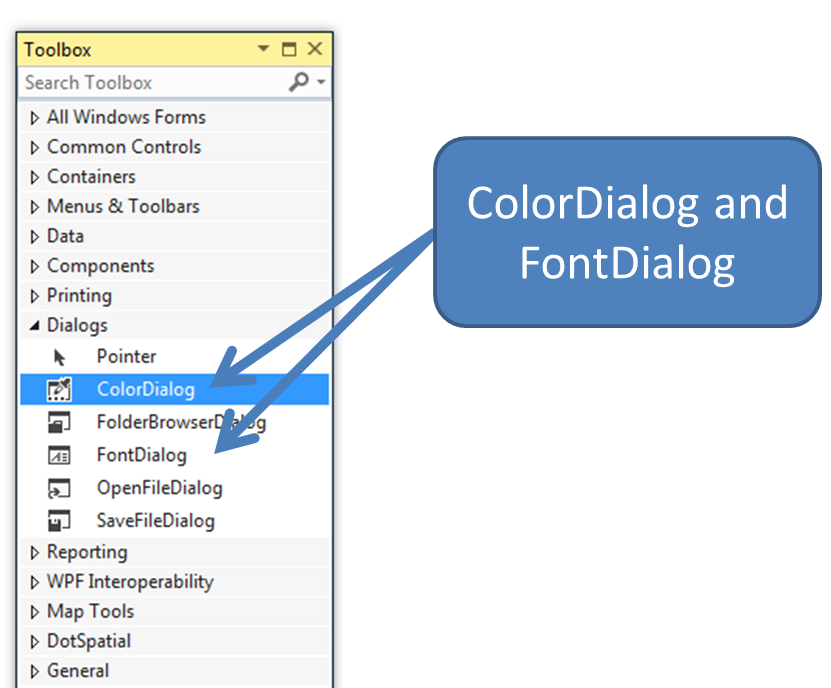


Figure 2: ColorDialog and FontDialog

Add a map control into the right SpatialDockManger panel and set its properties as follows:

Dock : Fill

Add three Groupboxes and set those properties as follows:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Properties | Groupbox1 | | Groupbox2 | Groupbox3 |
| Name | | gbCustom | gbShapeAttribute | gbCustomAttribute |
| Text | | Set the label properties | Display label from Attribute table | Custom Attributes for existing shape file |

Add 2 buttons into the gbCustom group box and set those buttons properties as follows:

Name : btnsetFont Text : Set Font Style and Size

Name : btnsetColor Text: Set Color

Add the following controls into the gbShapeAttribute group box.

Add a label control and set its properties as follows:

Name : lblFieldName Text: Fields

Add a comboBox and set its properties as follows:

Name: cmbFiledName

Add a button and set its properties as follows:

Name : btnDisplayLabel Text : Display Labels

Add the following controls into the gbCustomAttribute group box.

Add a label control and set its properties as follows:

Name : lblLabelName Text: Label name

Add a text box and set its properties as follows:

Name: txtCustomAttribute

Add 3 buttons and set those properties as follows:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Properties | Button1 | Button2 | | Button3 | |
| Name | btnAdd | | btnSave | | btnDelete |
| Text | Add | | Save | | Delete |

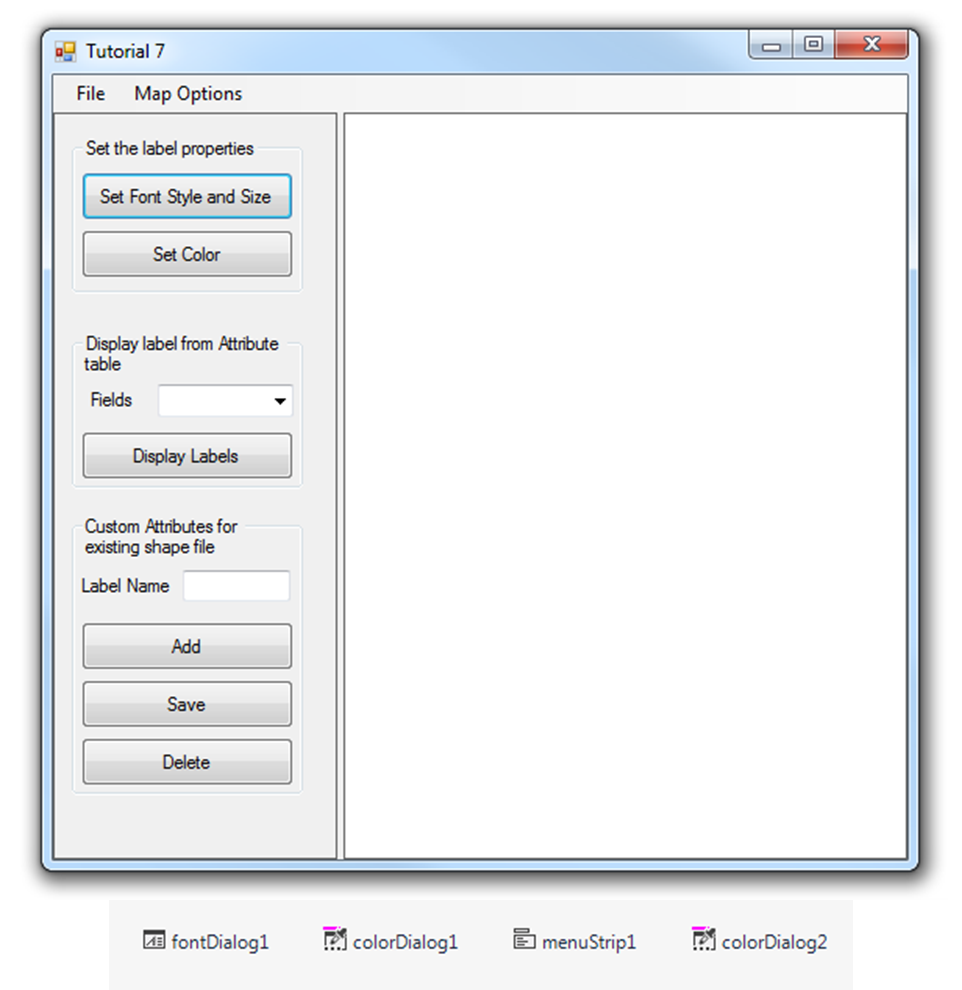


Figure 3: Final User Interface

**Step 4:** Code implementation

using DotSpatial.Data;

Write the following code in the Load menu item click event.

private void loadToolStripMenuItem\_Click(object sender, EventArgs e)

{

//Clear the existing layers from the map control

if ((map1.Layers.Count > 0))

{

map1.Layers.Clear();

}

//Clear the existing items from the combobox

cmbFiledName.Items.Clear();

//add the layers

OpenFileDialog fileDialog = new OpenFileDialog();

fileDialog.Filter = "Shapefiles|\*.shp";

if (fileDialog.ShowDialog() == DialogResult.OK)

{

//add layer to first map

IFeatureSet featureSet1 = FeatureSet.Open(fileDialog.FileName);

//FillColumnNames method is used to get all the attribute column names

//Based on the names combobox will be populated

FillColumnNames(featureSet1);

featureSet1.Reproject(map1.Projection);

map1.Layers.Add(featureSet1);

map1.ZoomToMaxExtent();

}

}

4.2 ) FillColumnNames method.

/// <summary>

/// This method is used to populate the attribute names from the shape file into the combobox

/// </summary>

/// <param name="featureSet"></param>

/// <remarks></remarks>

///

private void FillColumnNames(IFeatureSet featureSet)

{

foreach (DataColumn column in featureSet.DataTable.Columns)

{

cmbFiledName.Items.Add(column.ColumnName);

}

}

Display the label on the shape file from its attribute table.

Declare the following class level variables.

string fname = "Tahoma";

double fsize = 8.0;

Color fcolor = Color.Black;

Write the following code into the btnDisplayLabel\_Click event.

private void btnDisplayLabel\_Click(object sender, EventArgs e)

{

if ((cmbFiledName.Text == string.Empty))

{

MessageBox.Show("Please select an attribute from the textbox");

}

else

{

DisplayLabels(cmbFiledName.Text.ToString());

}

}

DisplayLabels method code.

/// <summary>

/// This method is used to display a lebel on a shape file base on user selection from combobox

/// </summary>

/// <param name="attributename">Attribute name</param>

/// <remarks></remarks>

private void DisplayLabels(string attributename)

{

//Check the number of layers from MapControl

if (map1.Layers.Count > 0)

{

map1.AddLabels((DotSpatial.Symbology.FeatureLayer)map1.Layers[0], "[" + attributename + "]", new Font("" + fname + "", (float)fsize), fcolor);

}

else

{

MessageBox.Show("Please add a layer to the map.");

}

}

**Display a custom label**

Write the following code btnAdd\_Click event.

private void btnAdd\_Click(object sender, EventArgs e)

{

DisplayCustomLabel(txtCustomAttribute.Text);

txtCustomAttribute.Text = "";

}

DisplayCustomLabel method code.

private void DisplayCustomLabel(string attributeValue)

{

if (string.IsNullOrEmpty(txtCustomAttribute.Text))

{

MessageBox.Show("Please enter the label text");

return;

}

IMapFeatureLayer selectedLayer = (IMapFeatureLayer)map1.Layers[0];

if (selectedLayer == null)

{

MessageBox.Show("Please add a layer to the map");

return;

}

int numSelectedFeatures = selectedLayer.Selection.Count;

if (numSelectedFeatures == 0)

{

MessageBox.Show("Please select a shape in the map");

return;

}

//create new column in attribute table

DataTable table = selectedLayer.DataSet.DataTable;

if (!(table.Columns.Contains("new\_label")))

{

table.Columns.Add(new DataColumn("new\_label"));

}

List<IFeature> selectedFeatureList = selectedLayer.Selection.ToFeatureList();

IFeature selectedFeature = selectedFeatureList[0];

selectedFeature.DataRow["new\_label"] = txtCustomAttribute.Text;

//display labels in the map

map1.AddLabels(selectedLayer, "[new\_label]", new Font("" + fname + "", (float)fsize), fcolor);

//reset map selection mode

map1.FunctionMode = FunctionMode.None;

}

Write the following code into the txtCustomAttribute\_TextChanged event.

private void txtCustomAttribute\_TextChanged\_1(object sender, EventArgs e)

{

map1.FunctionMode = FunctionMode.Select;

}

Write the following code into the btnSave\_Click event.

private void btnSave\_Click(object sender, EventArgs e)

{

if (map1.Layers.Count > 0)

{

IMapFeatureLayer stateLayer = default(IMapFeatureLayer);

stateLayer = (IMapFeatureLayer)map1.Layers[0];

stateLayer.DataSet.Save();

MessageBox.Show("Attribute has been saved in the shapefile.");

}

else

{

MessageBox.Show("Please add a layer to the map.");

}

}

Write the following code into the btnDelete\_Click event.

private void btnDelete\_Click(object sender, EventArgs e)

{

//Declare a datatable

DataTable dt = null;

if (map1.Layers.Count > 0)

{

IMapFeatureLayer stateLayer = default(IMapFeatureLayer);

stateLayer = (IMapFeatureLayer)map1.Layers[0];

//Get the shapefile's attribute table to our datatable dt

dt = stateLayer.DataSet.DataTable;

//Remove a column

dt.Columns.Remove("new\_label");

stateLayer.DataSet.Save();

MessageBox.Show("Attribute has been removed in the shapefile.");

}

else

{

MessageBox.Show("Please add a layer to the map.");

}

}

**Set label properties.**

**Set the font.**

Write the following code into the btnsetFont\_Click event.

private void btnSetFont\_Click(object sender, EventArgs e)

{

if (fontDialog1.ShowDialog() == System.Windows.Forms.DialogResult.OK)

{

fname = fontDialog1.Font.Name;

fsize = fontDialog1.Font.Size;

}

}

**Set the font color.**

Write the following code into the btnsetColor\_Click event.

private void btnsetColor\_Click(object sender, EventArgs e)

{

if (this.colorDialog1.ShowDialog() == DialogResult.OK)

{

fcolor = colorDialog1.Color;

}

}

**Menu strip controls code.**

Menu item s code.

private void clearToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.Layers.Clear();

}

private void exitToolStripMenuItem\_Click(object sender, EventArgs e)

{

this.Close();

}

private void zoomInToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.ZoomIn();

}

private void zoomOutToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.ZoomOut();

}

private void zoomToExtentToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.ZoomToMaxExtent();

}

private void panToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.FunctionMode = FunctionMode.Pan;

}

private void measureToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.FunctionMode = FunctionMode.Measure;

}

private void defaultToolStripMenuItem\_Click(object sender, EventArgs e)

{

map1.FunctionMode = FunctionMode.None;

}

***Output screen shot***

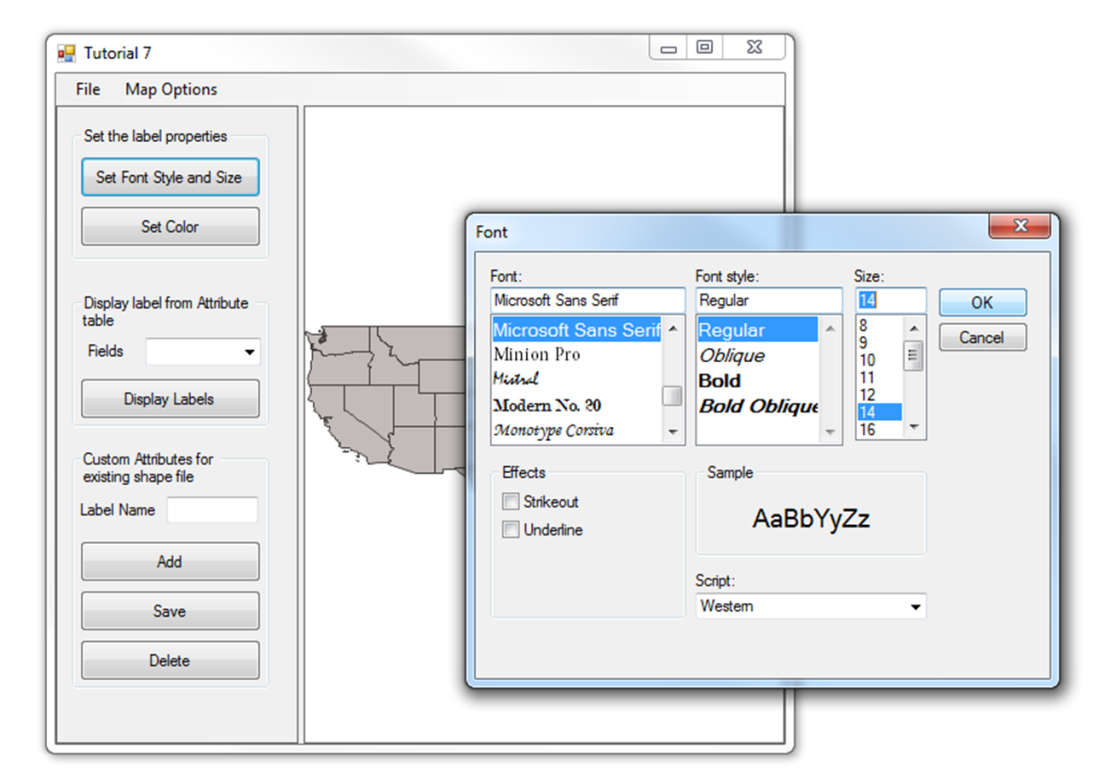


Figure 4: Set the Label Font Style and Size

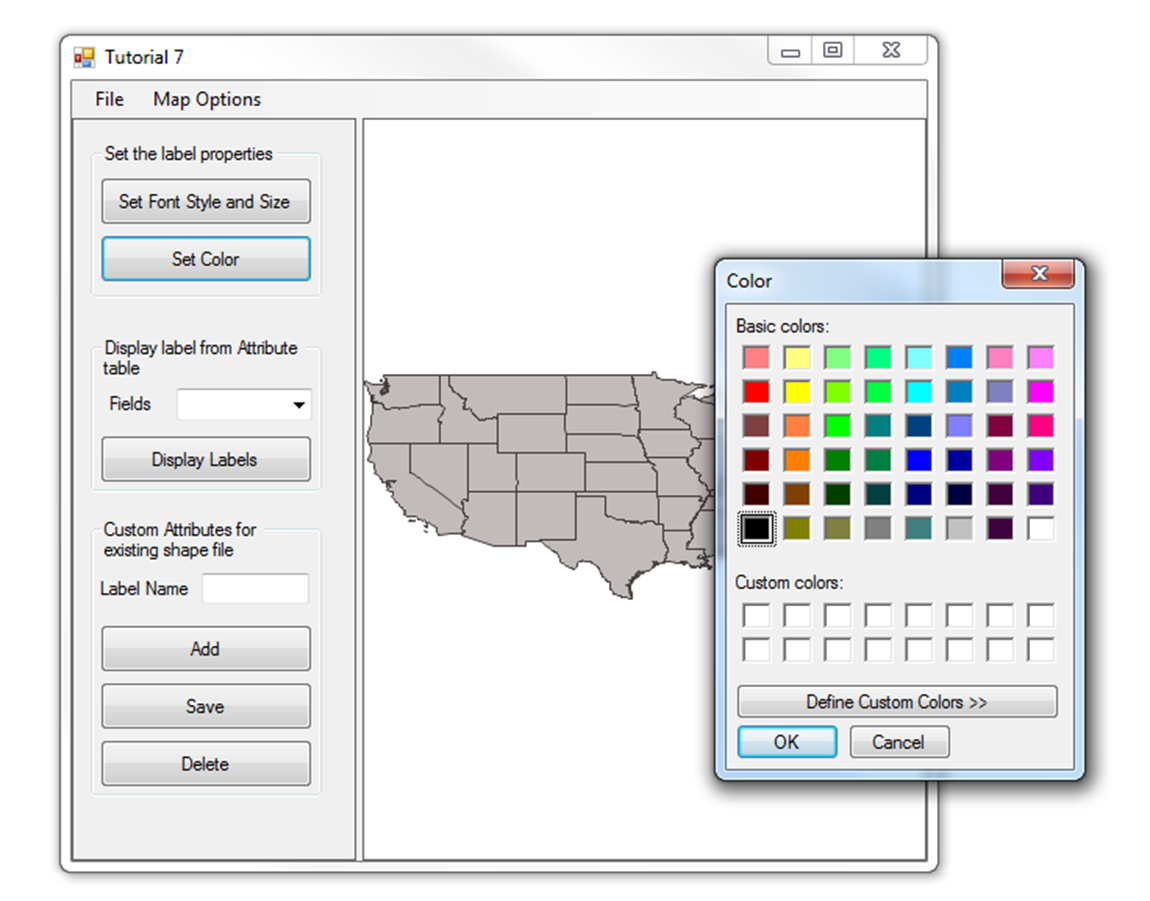


Figure 5: Set the Label Font Color

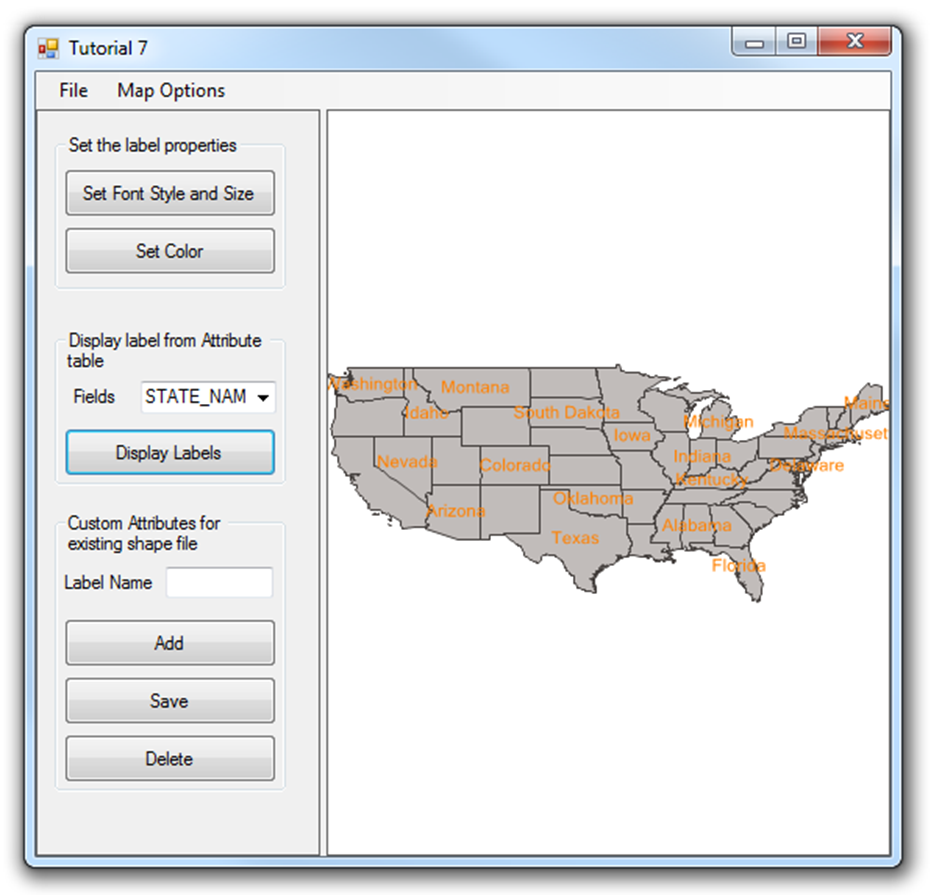


Figure 6: Displaying the Label

Be sure to click Display Labels each time you change the label’s properties in order to refresh the labels on the map.

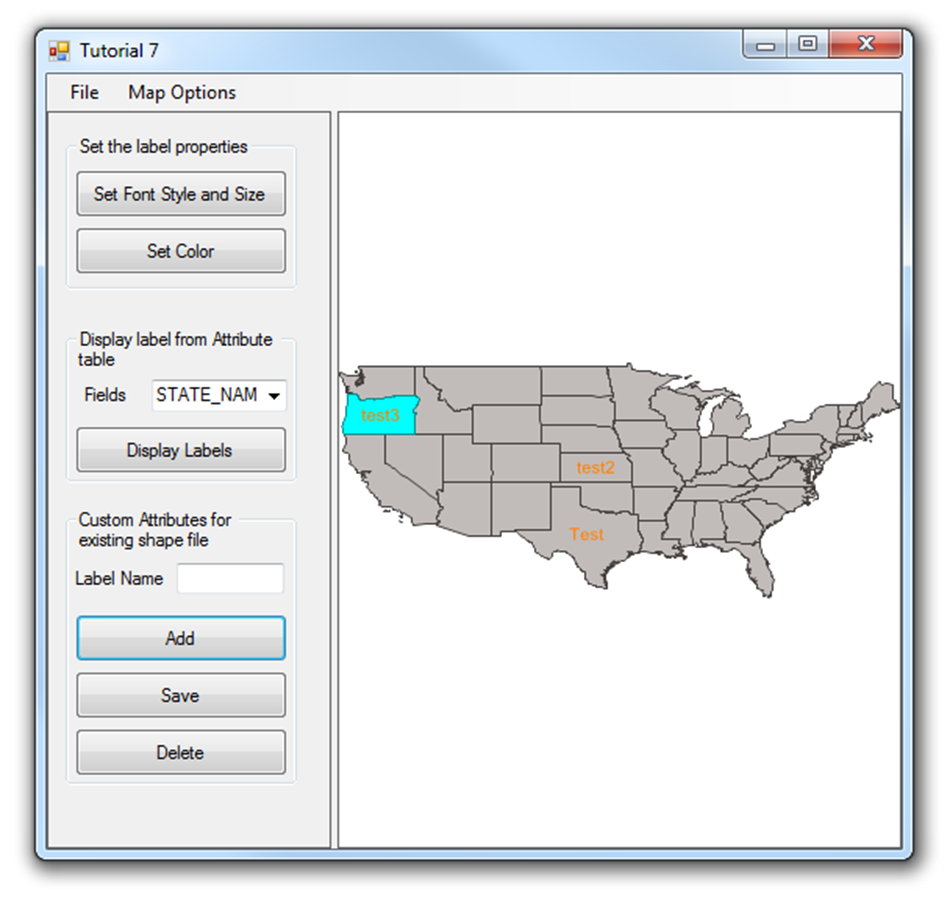


Figure 7: Create a Custom Label