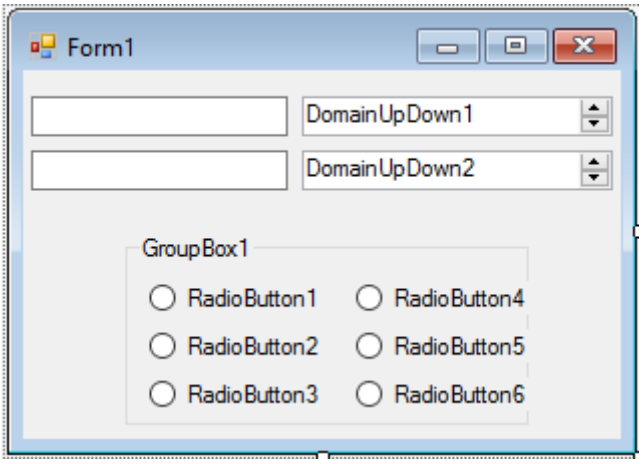
Project 6 – Units Conversion

Project Design

In this project, we will build a program that converts length from one unit of measure (inch, foot, yard, mile, centimeter, meter, kilometer) to another. The program will allow you to choose (using radio buttons) how many decimal points you want to display in the result. The project you are about to build is saved as **Conversion** in the project folder (**\BeginVCS\BVCS Projects**).

Place Controls on Form

Start a new project in Visual C#. Place two text box controls and two domain updown controls on the form. The domain updown controls are like a numeric updown, with items listed instead of numbers. Also, place a group box with six radio buttons. When done, your form should look something like this:



Set Control Properties

Set the control properties using the properties window:

Form1 Form:

Property Name	Property Value
Text	Units Conversion
BorderStyle	FixedSingle
StartPosition	CenterScreen

textBox1 Text Box:

Property Name	Property Value
Name	txtFromValue
Text	0
TextAlign	Right
Font Size	12

textBox2 Text Box:

Property Name	Property Value
Name	txtToValue
Text	0
TextAlign	Right
BackColor	White
Font Size	12
ReadOnly	True
TabStop	False

domainUpDown1 Domain UpDown:

Property Name	Property Value
Name	dudFromUnit
Text	[Blank]
TextAlign	Right
BackColor	Light Yellow
Font Size	12
ReadOnly	True

domainUpDown2 Domain UpDown:

Property Name	Property Value
Name	dudToUnit
Text	[Blank]
TextAlign	Right
BackColor	Light Yellow
Font Size	12
ReadOnly	True

groupBox1 Group Box:

Property Name	Property Value
Name	grpConvert
Text	Number of Decimals
Font Size	10

radioButton1 Radio Button:

Property Name	Property Value
Name	rdoDec0
Text	0
Checked	True

radioButton2 Radio Button:

Property Name	Property Value
Name	rdoDec1
Text	1

radioButton3 Radio Button:

Property Name	Property Value
Name	rdoDec2
Text	2

radioButton4 Radio Button:

Property Name	Property Value
Name	rdoDec3
Text	3

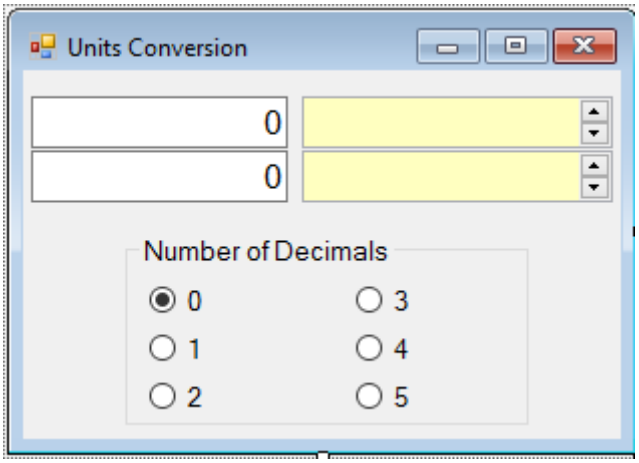
radioButton5 Radio Button:

Property Name	Property Value
Name	rdoDec4
Text	4

radioButton6 Radio Button:

Property Name	Property Value
Name	rdoDec5
Text	5

My finished form looks like this (resized some controls):



Write Event Methods

The idea of the program is simple. Type a value in the text box and choose the units (both From and To). With each change in value, units or number of decimals, the conversion is updated. Most of the computation is involved with the **txtFromValue_Change** event. Note how the conversion factors are stored in a two-dimensional array (table). Also note the use of the **NumberFormatInfo** object to specify the number of decimals to display. To use this object, you need these two lines after the left brace declaring the project **namespace** (right at the top of the code window):

```
using System;
using System.Globalization;
```

Add this code to the **general declarations** area:

```
string[] units = new string[7];
double [,] conversions = new double[7, 7];
bool loading = true;
NumberFormatInfo provider = new CultureInfo("en-US",
false).NumberFormat;
```

The **Form1_Load** event method:

```
private void Form1_Load(object sender, EventArgs e)
{
    int i;
    // Establish conversion factors - stored in two
dimensional array
    // or table - the first number is the table row, the
second number
    // the table column
    conversions[0, 0] = 1; //in to in
    conversions[0, 1] = 1.0 / 12; //in to ft
    conversions[0, 2] = 1.0 / 36; // in to yd
```

```
    conversions[0, 3] = (1.0 / 12) / 5280; // in to mi
    conversions[0, 4] = 2.54; // in to cm
    conversions[0, 5] = 2.54 / 100; // in to m
    conversions[0, 6] = 2.54 / 100000; // in to km
    for (i = 0; i < 7; i++)
    {
        conversions[1, i] = 12 * conversions[0, i];
        conversions[2, i] = 36 * conversions[0, i];
        conversions[3, i] = 5280 * (12 * conversions[0, i]);
        conversions[4, i] = conversions[0, i] / 2.54;
        conversions[5, i] = 100 * conversions[0, i] / 2.54;
        conversions[6, i] = 100000 * (conversions[0, i] /
2.54);
    }
    // Initialize variables
    units[0] = "inches (in)";
    units[1] = "feet (ft)";
    units[2] = "yards (yd)";
    units[3] = "miles (mi)";
    units[4] = "centimeters (cm)";
    units[5] = "meters (m)";
    units[6] = "kilometers (km)";
    for (i = 0; i < 7; i++)
    {
        dudFromUnit.Items.Add(units[i]);
        dudToUnit.Items.Add(units[i]);
    }
    dudFromUnit.SelectedIndex = 0;
    dudToUnit.SelectedIndex = 0;
    provider.NumberDecimalDigits = 0;
    // Put cursor in text box
    txtFromValue.Focus();
    loading = false;
}
```

The **txtFromValue_KeyPress** event method:

```
private void txtFromValue_KeyPress(object sender,
KeyPressEventArgs e)
{
    // Numbers and decimal point only
    if ((e.KeyChar >= '0' && e.KeyChar <= '9') || e.KeyChar
== '.' || (int) e.KeyChar == 8)
    {
        e.Handled = false;
    }
    else
    {
        e.Handled = true;
    }
}
```

The **txtFromValue_TextChanged** event method:

```
private void txtFromValue_TextChanged(object sender,
EventArgs e)
{
    UpdateDisplay();
}
```

The **UpdateDisplay** general method (recall type the method after any other method):

```
private void UpdateDisplay()
{
    if (loading)
    {
        return;
    }
    double v;
    // Do unit conversion
    v = conversions[dudFromUnit.SelectedIndex,
dudToUnit.SelectedIndex] *
Convert.ToDouble(txtFromValue.Text);
    txtToValue.Text = v.ToString("N", provider);
    txtFromValue.Focus();
}
```

The **dudFromUnit_SelectedItemChanged** event method:

```
private void dudFromUnit_SelectedItemChanged(object sender,
EventArgs e)
{
    UpdateDisplay();
}
```

The **dudToUnit_SelectedItemChanged** event method:

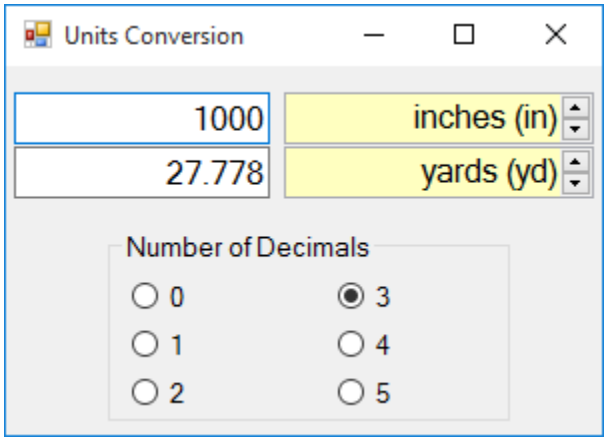
```
private void dudToUnit_SelectedItemChanged(object sender,
EventArgs e)
{
    UpdateDisplay();
}
```

The **rdoDec0_CheckedChanged** to **rdoDec5_CheckedChanged** event methods:

```
private void rdoDec0_CheckedChanged(object sender, EventArgs
e)
{
    provider.NumberDecimalDigits = 0;
    UpdateDisplay();
}

private void rdoDec1_CheckedChanged(object sender, EventArgs
e)
{
    provider.NumberDecimalDigits = 1;
    UpdateDisplay();
}

private void rdoDec2_CheckedChanged(object sender, EventArgs
e)
{
    provider.NumberDecimalDigits = 2;
    UpdateDisplay();
}

private void rdoDec3_CheckedChanged(object sender, EventArgs
e)
{
    provider.NumberDecimalDigits = 3;
    UpdateDisplay();
}

private void rdoDec4_CheckedChanged(object sender, EventArgs
e)
{
    provider.NumberDecimalDigits = 4;
    UpdateDisplay();
}

private void rdoDec5_CheckedChanged(object sender, EventArgs
e)
{
    provider.NumberDecimalDigits = 5;
    UpdateDisplay();
}
```

Run the Project

Save your work. Run the project. Type in a value. Watch the corresponding converted value change as you type. Change the From units and the To units using the updown controls. Change the number of decimals. Make sure all the options work as designed. Here's a run I tried:



Other Things to Try

The most obvious change to this program is to include other units of measure. You could build a general purpose units conversion program that converts not only length, but weight, volume, density, area, temperature and many others. Such a program would be invaluable.