

Drawing

Introduction

- ❖ We can use simple drawing in the process of creating new controls.
- ❖ Through the drawing itself we can use various controls such as Eclipse, Line, Path, Polygon, Polyline and Rectangle.

The Stroke Properties

- ❖ Through the stroke properties we can control the way our lines are drawn.
- ❖ The `Fill` property is the brush in use when drawing the background.
- ❖ The `Stroke` property is the brush in use for drawing the edges.

The Stroke Properties

- ❖ The `StrokeDashArray` property holds an array of values that specify the number of pixels to be drawn and then be skipped. These numeric values are scaled in accordance with the line's thickness.
- ❖ The `StrokeDashCap` property determines the ends of dashes shape. The possible values are `Flat`, `Round`, `Square` or `Triangle`.

The Stroke Properties

- ❖ The `StrokeDashOffset` determines the distance from the beginning of the line to the first dash.
- ❖ The `StrokeEndLineCap` determines the shape of the end of the line. That shape can be `Flat`, `Square`, `Round` or `Triangle`.
- ❖ The `StrokeLineJoin` determines how to join adjacent lines in shapes such as rectangles, polygons and polylines. The possible values are `Miter`, `Round` or `Bevel`.

The Stroke Properties

- ❖ The `StrokeStartLineCap` property determines the shape the line should start with.
- ❖ The `StrokeThickness` property determines the width of the line.

The Stroke Properties

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Simple Stroke Properties Demo"
        Width="400" Height="300" Background="White">

    <WrapPanel Height="100" Width="200">

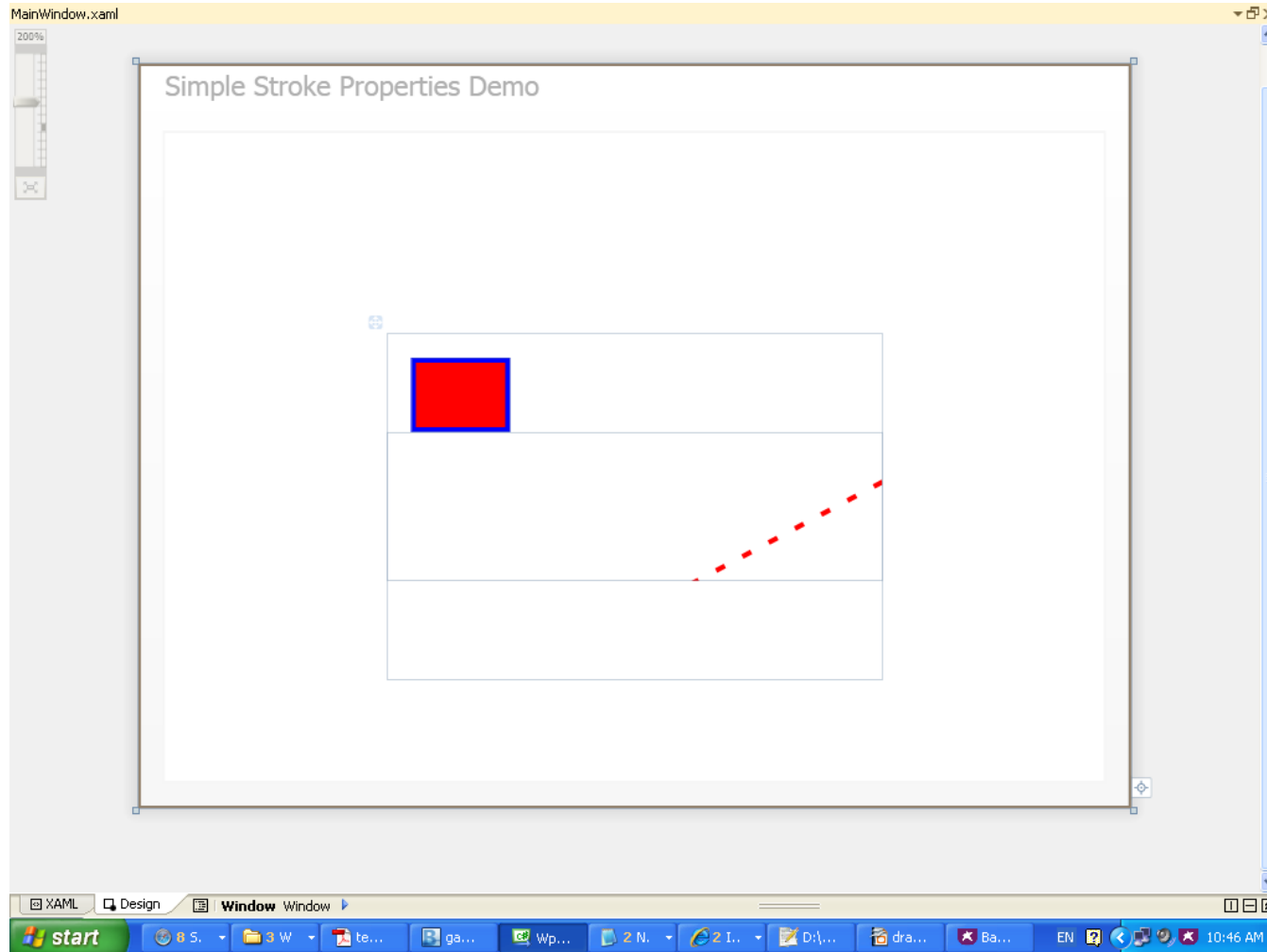
        <Rectangle Width="40" Height="30"
                   Stroke="Blue" StrokeThickness="2"
                   Fill="Red" HorizontalAlignment="Left"
                   VerticalAlignment="Top"
                   Margin="10,10,0,0"/>

        <Line X1="200" Y1="20" X2="50" Y2="100"
              Stroke="Red" StrokeThickness="2"
              StrokeDashArray="2,4"/>

    </WrapPanel>

</Window>
```

The Stroke Properties



The Ellipse Control

- ❖ We can use this control for drawing simple ellipses that don't contain any child elements.
- ❖ This control provides us with events we can use for handling user interaction events related to this control. The more important of these events are `MouseDown`, `MouseEnter` and `MouseLeave`.

The Ellipse Control

- ❖ The `Width` and the `Height` properties set the size. When omitting the `Width` and the `Height` properties the ellipse re-sizes together with its container. The location is set by the container.

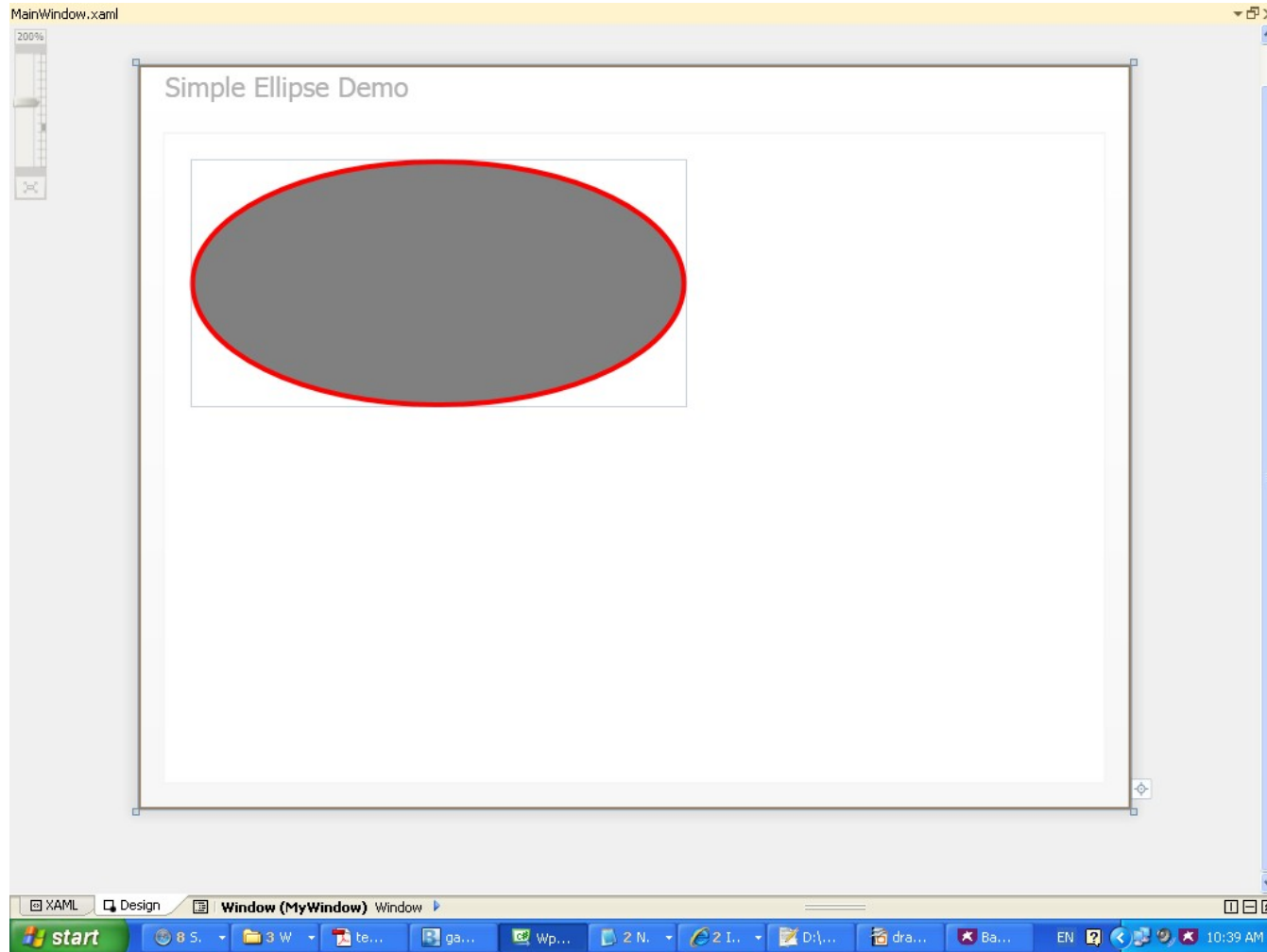
The Ellipse Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  x:Name="MyWindow"
  Title="Simple Ellipse Demo"
  Width="400" Height="300" Background="White">

  <Ellipse Width="200" Height="100"
    Stroke="Red" StrokeThickness="2"
    Fill="Gray" HorizontalAlignment="Left"
    VerticalAlignment="Top"
    Margin="10,10,0,0"/>

</Window>
```

The Ellipse Control



The Line Control

- ❖ This control draws a line segment. The location is determined using its `X1`, `Y1`, `X2` and `Y2` properties.
- ❖ This control provides a set of events through which we can handle the user interaction with the line we get.

The Line Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Simple Line Demo"
        Width="600" Height="400" Background="White">

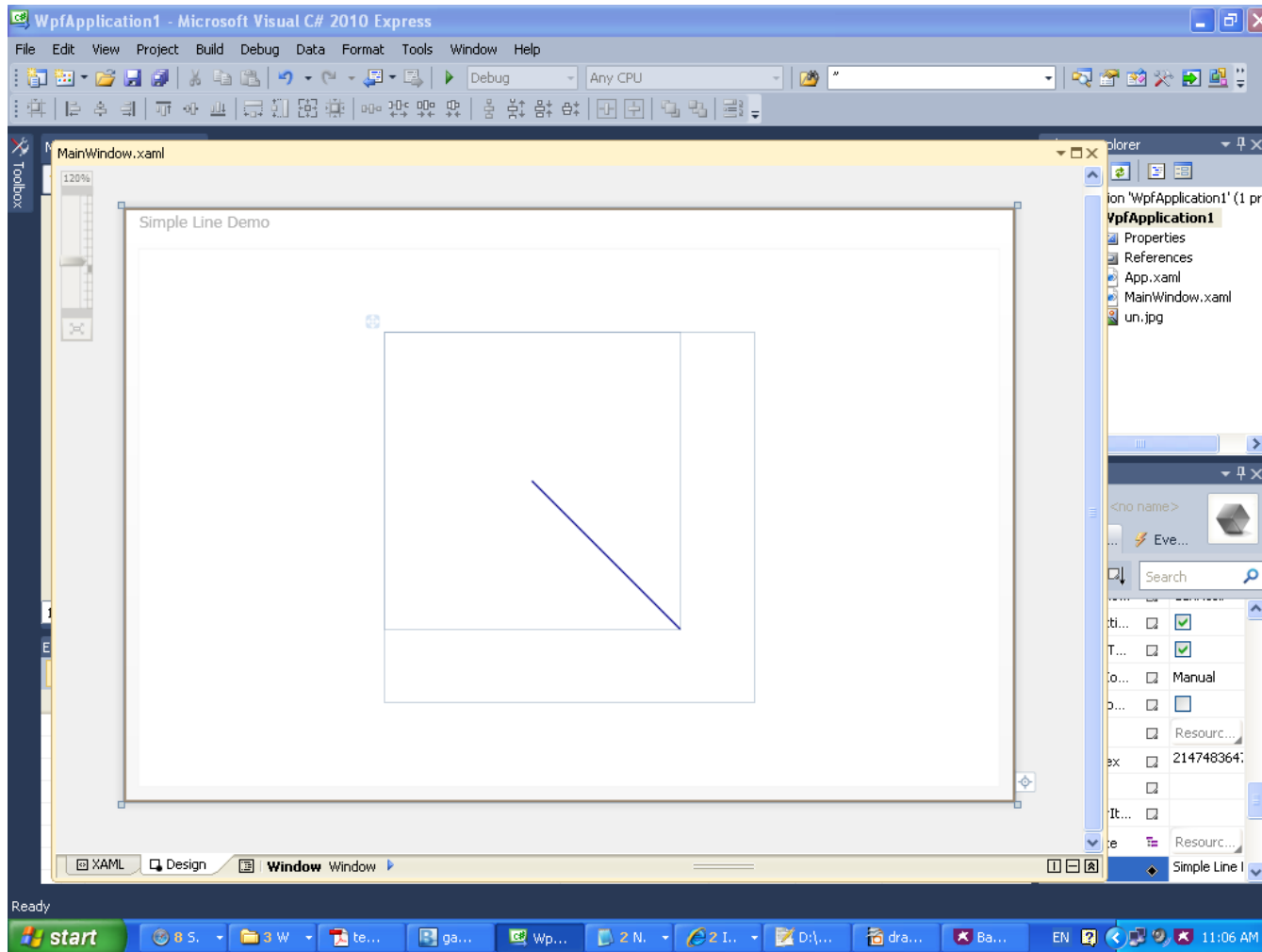
    <WrapPanel Height="250" Width="250">

        <Line X1="100" Y1="100" X2="200" Y2="200"
              Stroke="DarkBlue"/>

    </WrapPanel>

</Window>
```

The Line Control



The Path Control

- ❖ This control draws a path composed of series of small shapes such as lines, arcs, curves and others.
- ❖ We use the `Path` mini language when creating paths using this control.

The Path Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Simple Path Demo" Width="600" Height="400"
  Background="White">

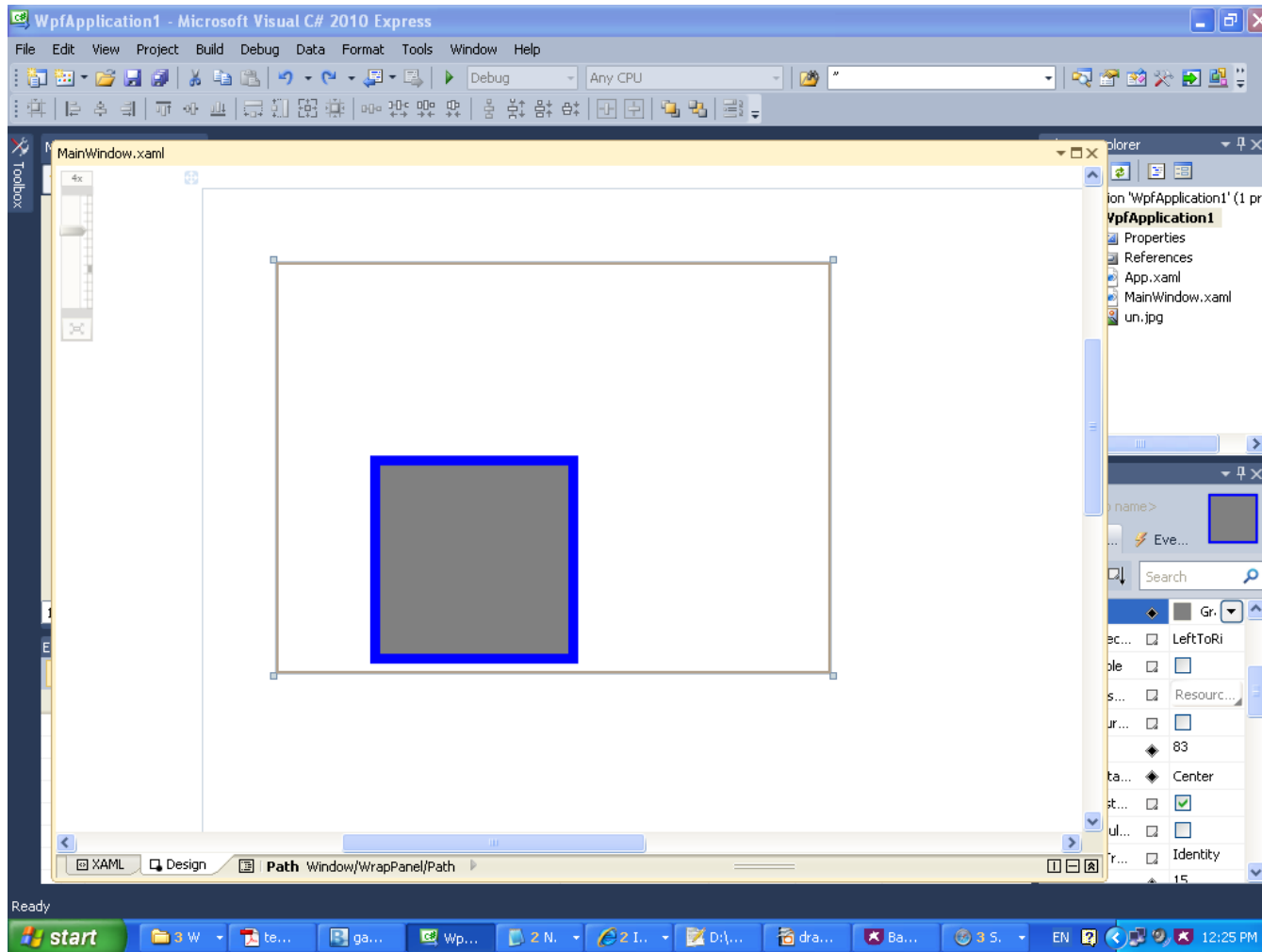
  <WrapPanel Height="250" Width="250">

    <Path Margin="15" HorizontalAlignment="Center"
      VerticalAlignment="Top" Width="112" Height="83"
      Fill="Gray" Stroke="Blue" StrokeThickness="2"
      Data="M20,40 L20,80 L60,80 L60,40 Z"/>
  </WrapPanel>

</Window>
```

Path Mini Language

The Path Control



The Path Mini Language

- ❖ Some of the Path mini language commands are followed by one or more points. These are parameters the commands use.
- ❖ We separate the points either by placing spaces.
- ❖ When the command is in upper case then it means that the points are absolute.
- ❖ When the command is in lower case then it means that the points are relative.

The Path Mini Language

| command | explanation |
|---------|---|
| F0 | use the odd even fill rule |
| F1 | use the nonzero fill rule |
| M or m | move to specific point |
| L or l | draw line to specific point |
| H or h | draw horizontal line to specific point |
| V or v | draw vertical line to specific point |
| C or c | draw a cubic curve |
| S or s | draw a bezier curve |
| Q or q | draw a quadratic bezier curve |
| T or t | draw a bezier curve defined by one point only |
| A or a | draw elliptical arc |
| Z or z | close the path by drawing a line to the first point |

Cubic Curve

- ❖ The cubic curve command takes three points. Two of them are the control points. The third is the end point.
- ❖ The curve starts at the current point and moves toward the first control point and then moves away from the second control point and ends at the end point.

Cubic Curve

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Simple Line Demo" Width="400" Height="300" Background="White">

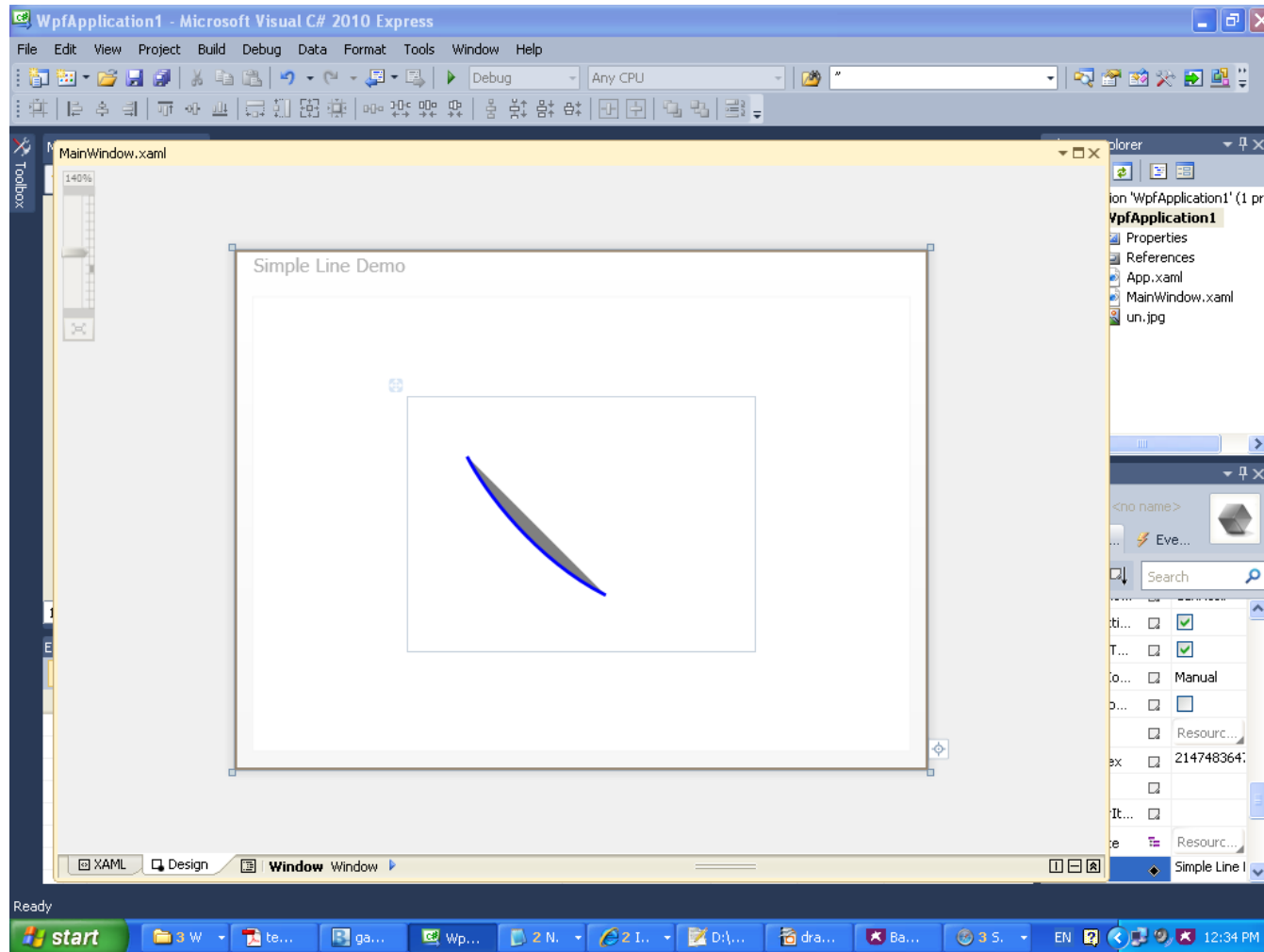
    <WrapPanel Height="148" Width="202">

        <Path Margin="15" HorizontalAlignment="Center"
            VerticalAlignment="Top" Width="159" Height="123"
            Fill="Gray" Stroke="Blue" StrokeThickness="2"
            Data="M20,20 C30,40 60,80 100,100"/>

    </WrapPanel>

</Window>
```

Cubic Curve



Bezier Curve

- ❖ The bezier curve S command takes two points. The first is the control point. The second is the end point.

Bezier Curve

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Bezier Curve Demo" Width="400" Height="300"
        Background="White">

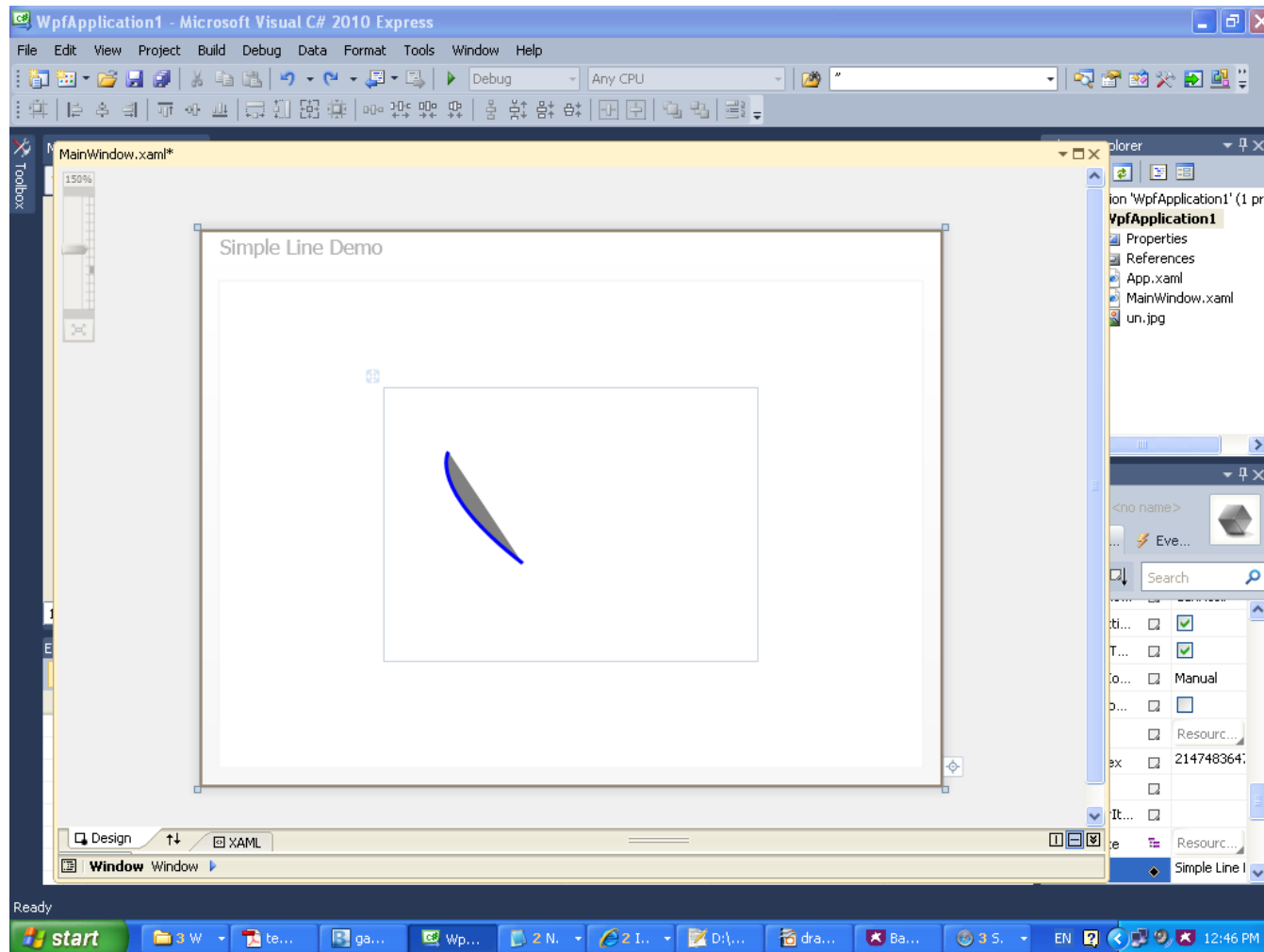
    <WrapPanel Height="148" Width="202">

        <Path Margin="15" HorizontalAlignment="Center"
              VerticalAlignment="Top" Width="159" Height="123"
              Fill="Gray" Stroke="Blue" StrokeThickness="2"
              Data="M20,20 S8,40 60,80"/>

    </WrapPanel>

</Window>
```

Bezier Curve



Quadratic Bezier Curve

- ❖ The quadratic bezier curve Q command takes two points. The first is the control point. The second is the endpoint.
- ❖ The curve starts at the current point, moves towards the control point and ends at the endpoint while moving away from the control point.

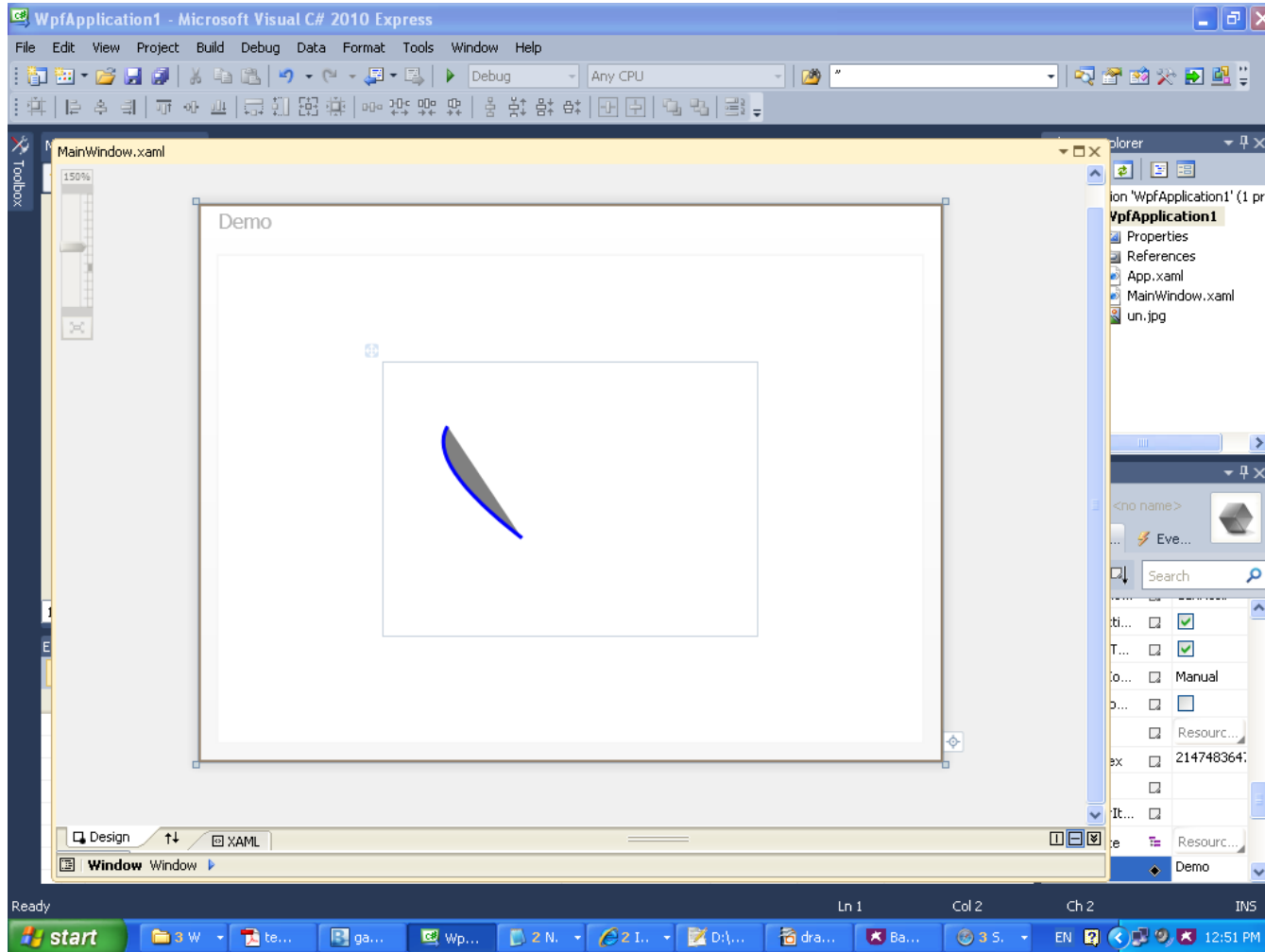
Quadratic Bezier Curve

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Demo"
        Width="400" Height="300" Background="White">

    <WrapPanel Height="148" Width="202">
        <Path Margin="15" HorizontalAlignment="Center"
              VerticalAlignment="Top" Width="159" Height="123"
              Fill="Gray" Stroke="Blue" StrokeThickness="2"
              Data="M20,20 Q8,40 60,80"/>
    </WrapPanel>

</Window>
```

Quadratic Bezier Curve



Bezier Curve with One Point

- ❖ The T command creates a bezier curve defined by a single point.

Bezier Curve with One Point

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
  Title="Demo" Width="400" Height="300" Background="White">

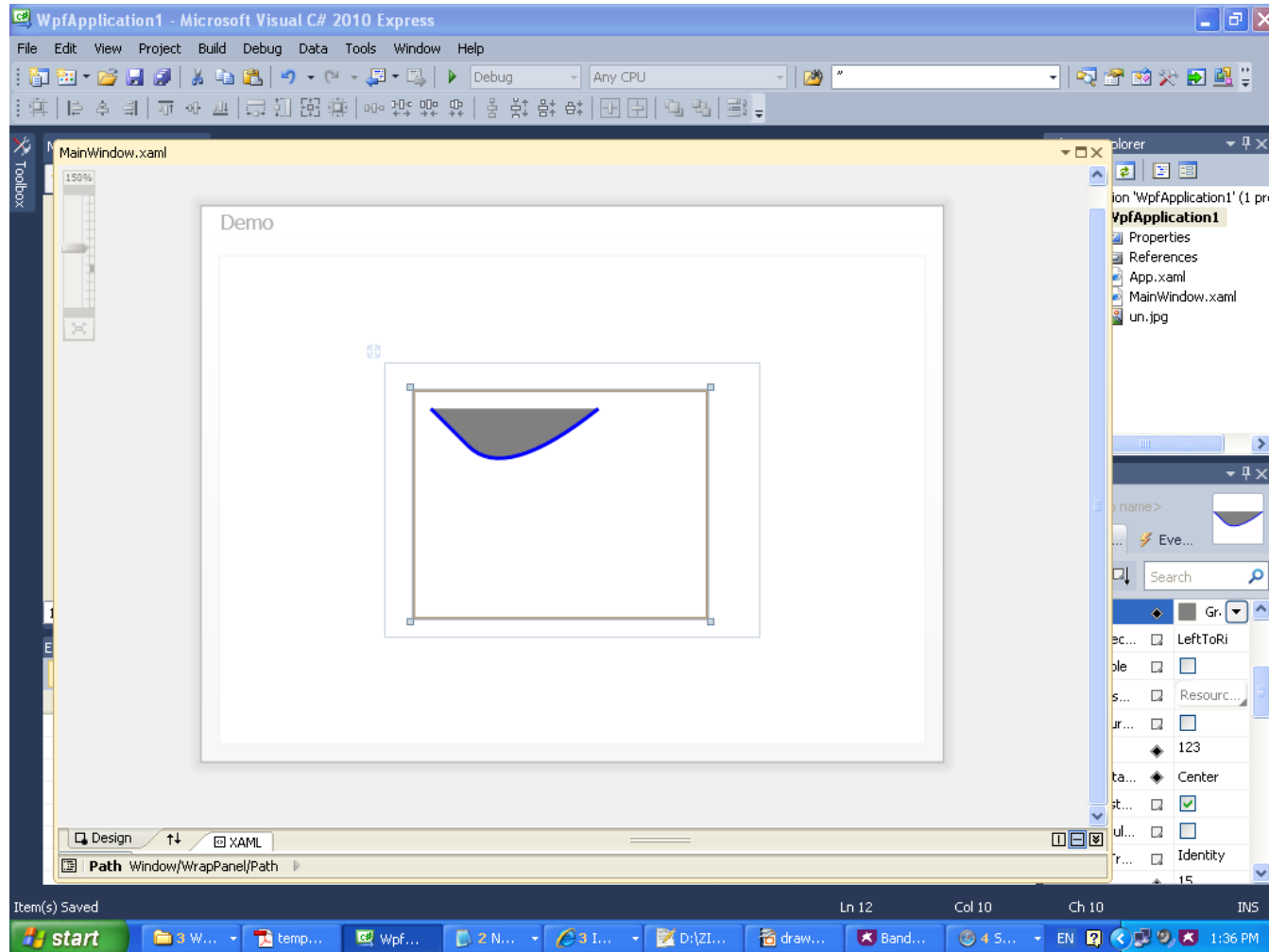
  <WrapPanel Height="148" Width="202">

    <Path Margin="15" HorizontalAlignment="Center"
      VerticalAlignment="Top" Width="159" Height="123"
      Fill="Gray" Stroke="Blue" StrokeThickness="2"
      Data="M10,10 T30,30 100,10"/>

  </WrapPanel>

</Window>
```

Bezier Curve with One Point



Arcs

❖ The `A` command creates an arc that starts at the current point.

❖ This command uses five parameters.

`size` – x and y that defines the radius of the arc

`rotation_angle` – angle of rotation in degrees

`large_angle` – if 0 then the arc will span less than 180 degrees. if 1 then the arc will span 180 degrees.

`sweep_direction` – if 0 the direction changes counterclockwise. if 1 the direction changes clockwise

`end_point` – where the arc ends

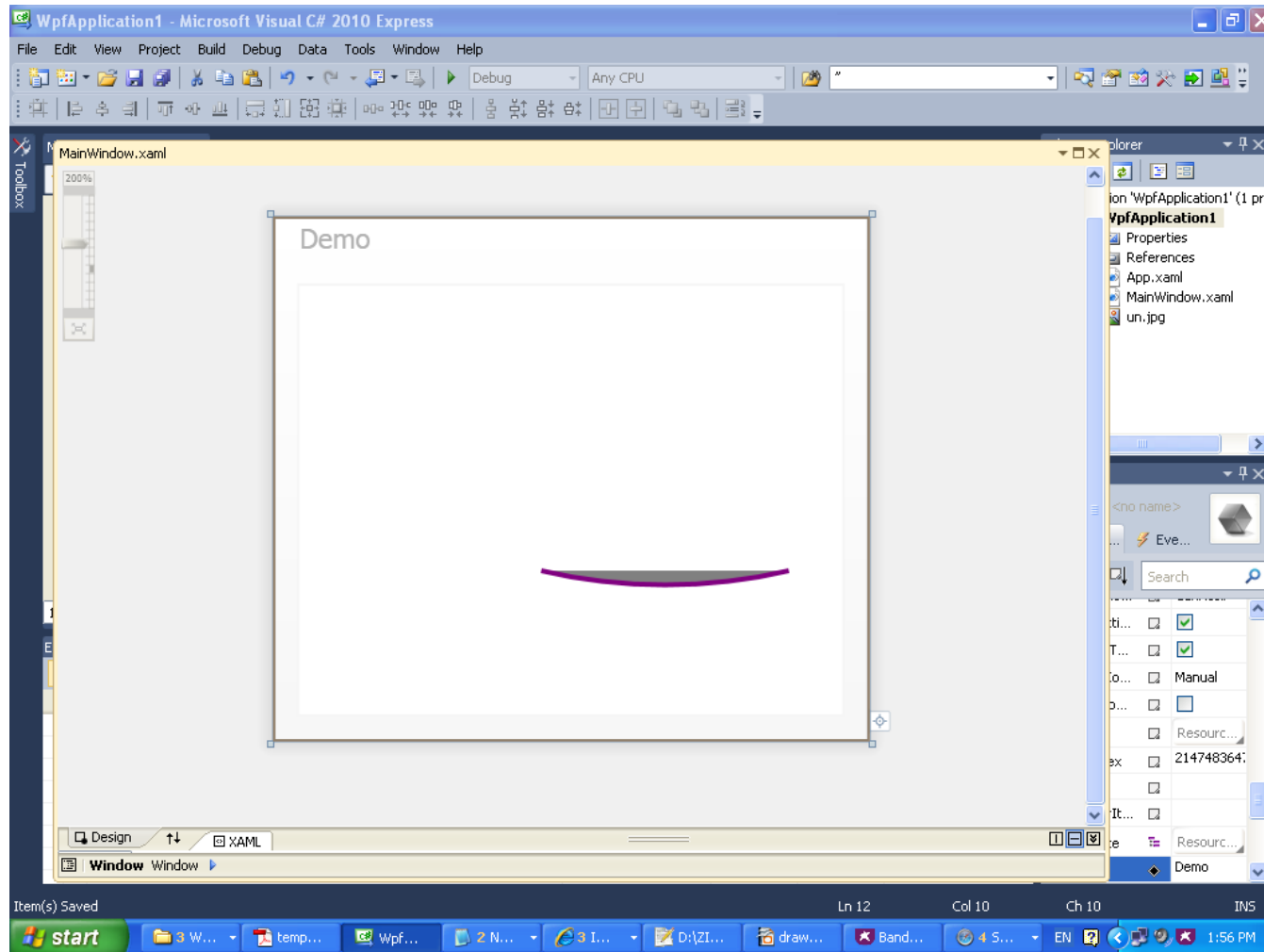
Arcs

```
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Demo"
Width="241" Height="212" Background="White">

    <Path Margin="15" HorizontalAlignment="Center"
VerticalAlignment="Top" Width="224" Height="150"
Fill="Gray" Stroke="Purple" StrokeThickness="2"
Data="M100,100 A150,100 180 0 0 200,100"/>

</Window>
```

Arcs



Path of Objects

- ❖ Instead of using the path mini language we can add child objects to our path object

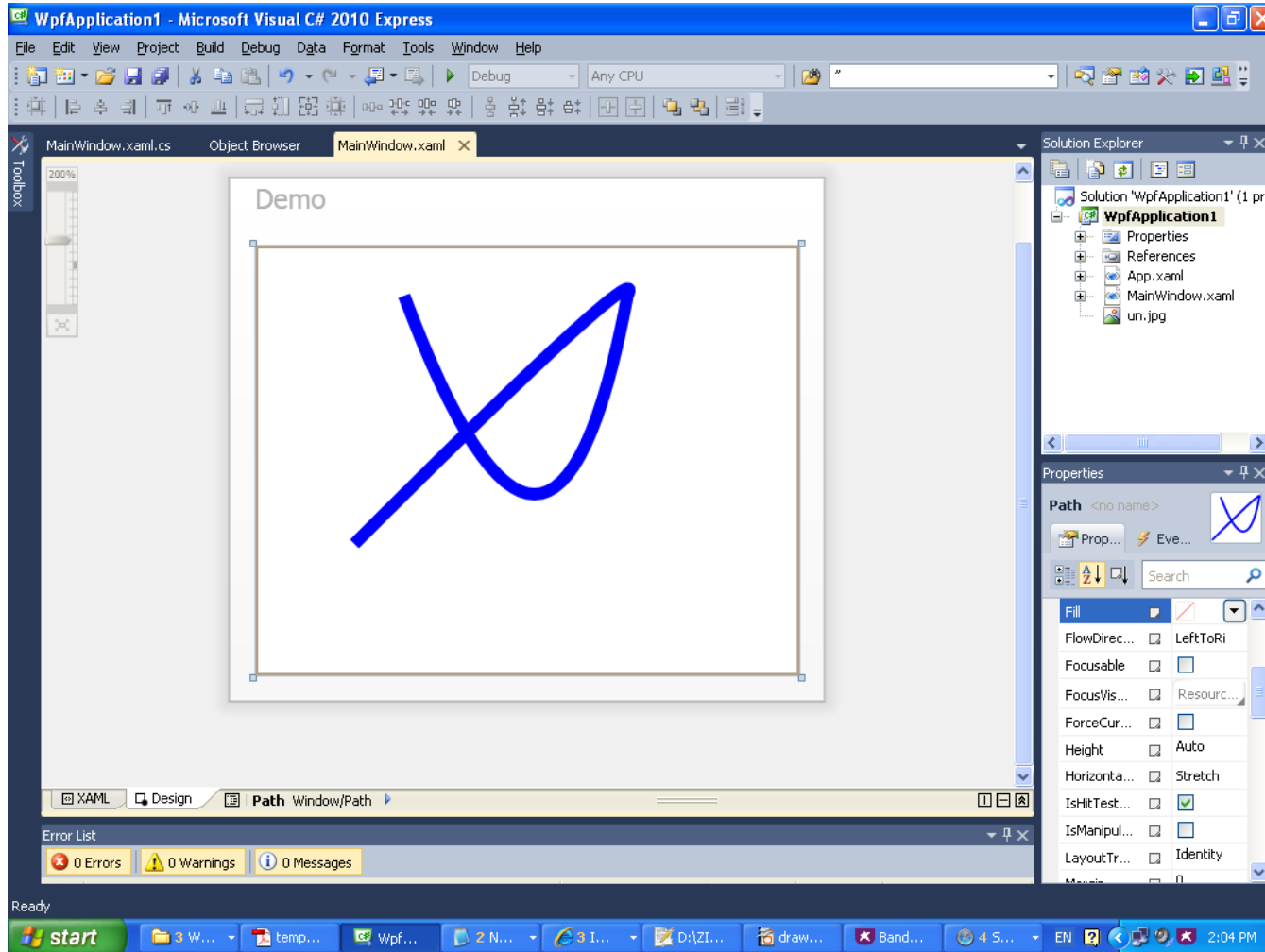
Path of Objects

```
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Demo"
Width="241" Height="212" Background="White">

    <Path Stroke="Blue" StrokeThickness="5">
        <Path.Data>
            <PathGeometry>
                <PathGeometry.Figures>
                    <PathFigureCollection>
                        <PathFigure StartPoint="60,20">
                            <PathFigure.Segments>
                                <PathSegmentCollection>
                                    <PolyQuadraticBezierSegment
Points="120,180 150,20 160,00 40,120 40,40"/>
                                </PathSegmentCollection>
                            </PathFigure.Segments>
                        </PathFigure>
                    </PathFigureCollection>
                </PathGeometry.Figures>
            </PathGeometry>
        </Path.Data>
    </Path>

</Window>
```

Path of Objects



The Polygon Control

- ❖ This object draws lines segments that connect a series of points.
- ❖ The last point is automatically connected with the first.

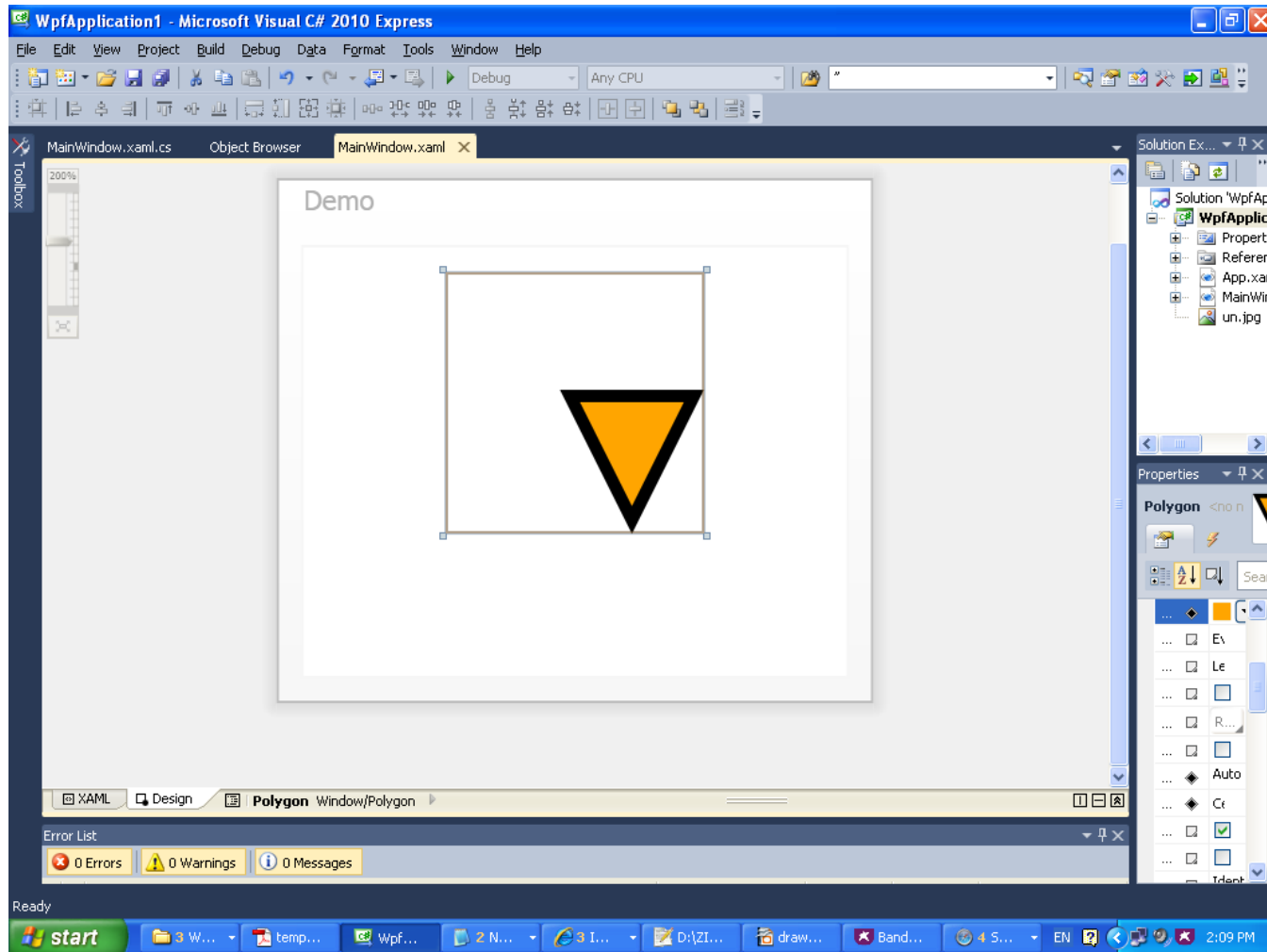
The Polygon Control

```
<Window
xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
Title="Demo" Width="241" Height="212" Background="White">

    <Polygon      Margin="10" HorizontalAlignment="Center"
                  VerticalAlignment="Top" Width="Auto"
                  Height="Auto" Fill="Orange"
                  Stroke="#FF000000" StrokeThickness="5"
                  Points="50,50 100,50, 75,100"/>

</Window>
```


The Polygon Control



The Polyline Control

- ❖ This object draws line segments that connect the series of points we pass over to it.
- ❖ Unlike the `Polygon` object it doesn't automatically connect the last point to the first.

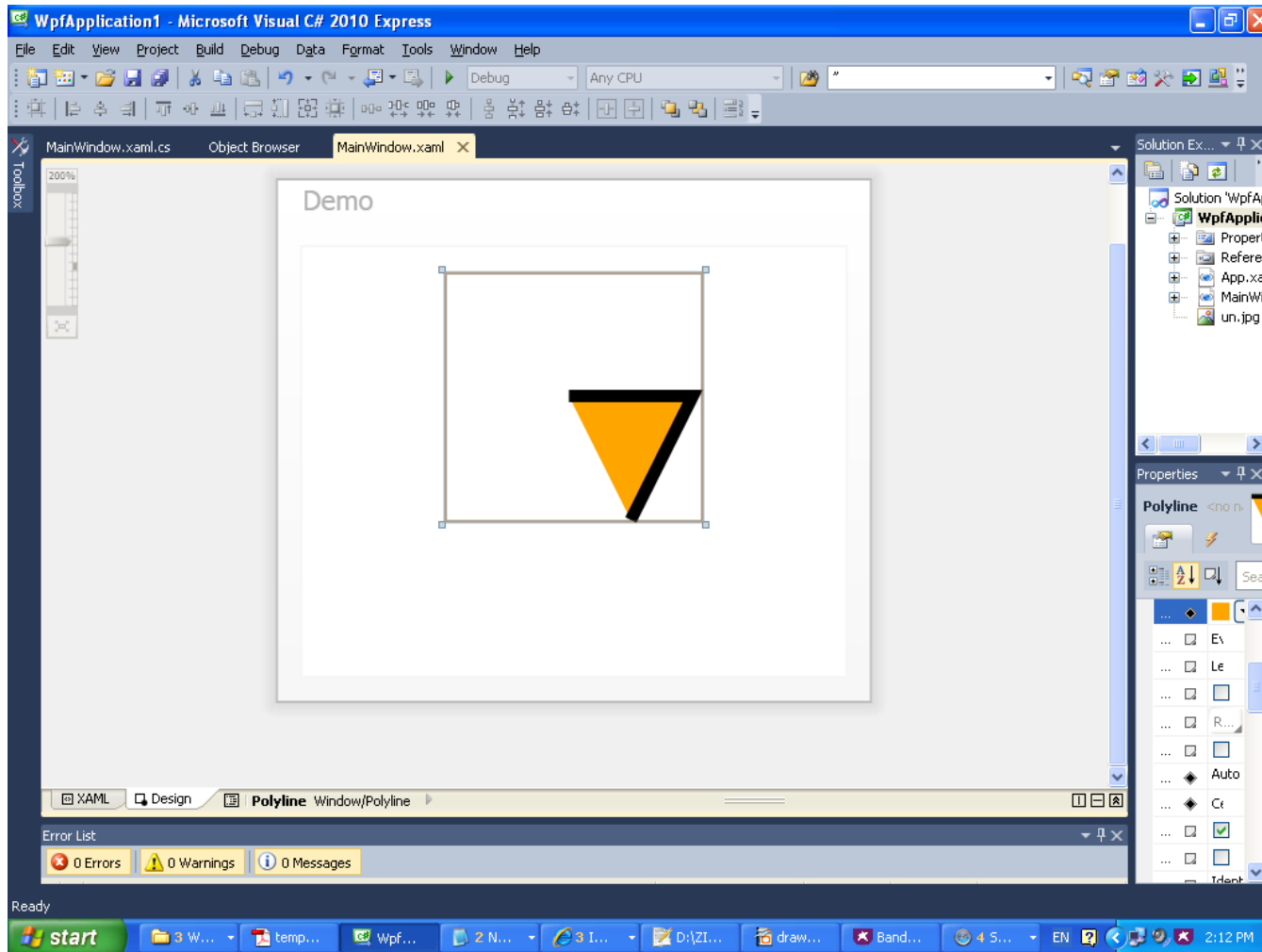
The Polyline Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Demo" Width="241" Height="212" Background="White">

    <Polyline Margin="10" HorizontalAlignment="Center"
              VerticalAlignment="Top" Width="Auto"
              Height="Auto" Fill="Orange"
              Stroke="#FF000000" StrokeThickness="5"
              Points="50,50 100,50, 75,100"/>

</Window>
```

The Polyline Control



The Rectangle Control

- ❖ This object draws a simple rectangle that cannot contain any children.
- ❖ The rectangle size is determined by its Width and Height. The location is determined by the container.

The Rectangle Control

```
<Window xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="Demo" Width="241" Height="212" Background="White">

    <Rectangle Width="200" Height="100" Margin="10"
        Stroke="Red" StrokeThickness="10"/>

</Window>
```

The Rectangle Control

