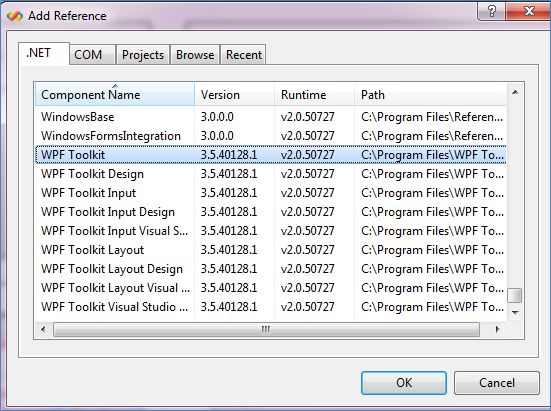
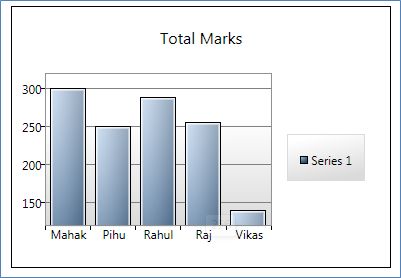
WPF Chart Control:-

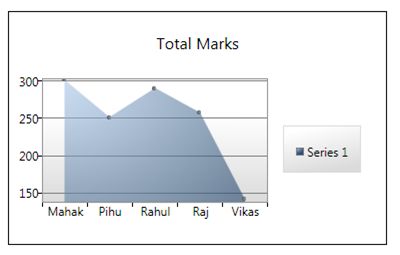
In this article we will discuss the many types of Chart Controls In WPF. Here we discuss the Column, Line, Bar, Pie and Area Charts.  
So First we create the Column Chart.  
  
**Column Chart:** Here we take a simple example of the total marks of the students in the class. For this follow these steps:  
  
**Step 1:** Download the WPF Toolkit from the following link:  
  
<http://wpf.codeplex.com/>  
  
**Step 2:** Now add the Reference to your project as:  
  
**Project menu:-> Add Reference**  
  
  
  
**Step 3:** Now add the following in the <window> tag:  
  
<Window x:Class="Chart\_Control\_In\_WPF.Window1"  
    xmlns="<http://schemas.microsoft.com/winfx/2006/xaml/presentation>"  
    xmlns:x="<http://schemas.microsoft.com/winfx/2006/xaml>"  
    Title="Window1" Height="916" Width="1137" **xmlns:my="clr-namespace:System.Windows.Controls.DataVisualization.Charting;assembly=  
System.Windows.Controls.DataVisualization.Toolkit**">  
    <ScrollViewer HorizontalScrollBarVisibility="Auto"  
      VerticalScrollBarVisibility="Auto" Margin="0,-28,0,28">  
<Window>  
  
**Step 4:**Now add the following code in the grid to add the chart control:  
  
<Grid Height="921">  
            <my:Chart Height="262" HorizontalAlignment="Left"   
            Margin="33,0,0,620" Name="ColumnChart1" Title="Total Marks"  
            VerticalAlignment="Bottom" Width="380">  
                <my:ColumnSeries DependentValuePath="Value"

            </my:Chart>  
        </Grid>  
  
**Step 5:** After that we add the code in the .cs file:  
  
public Window1()  
{  
    InitializeComponent();  
    showColumnChart();

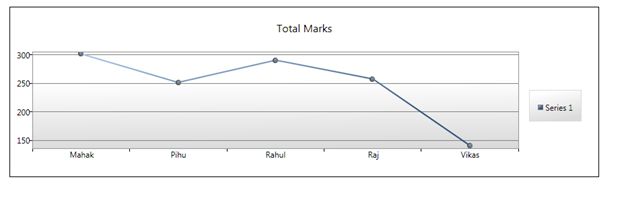
}  
private void showColumnChart()  
{  
    List<KeyValuePair<string, int>> MyValue = new List<KeyValuePair<string, int>>();  
    MyValue.Add(new KeyValuePair<string, int>("Mahak", 300));  
    MyValue.Add(new KeyValuePair<string, int>("Pihu", 250));  
    MyValue.Add(new KeyValuePair<string, int>("Rahul", 289));  
    MyValue.Add(new KeyValuePair<string, int>("Raj", 256));  
    MyValue.Add(new KeyValuePair<string, int>("Vikas", 140));

    ColumnChart1.DataContext = MyValue;

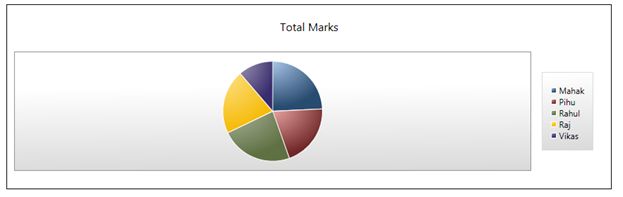
}  
  
Here we call the function showColumnChart(). In which we provide the values we want to show in the chart. Here we take the KeyValuePair which will be used to show the value in the chart like this:  
  
<my:Chart  Name="AreaChart1" Title="Total Marks" Margin="33,330,0,358" HorizontalAlignment="Left" Width="379">  
                <my:**AreaSeries DependentValuePath="Value"  
          IndependentValuePath="Key"** ItemsSource="{Binding}"  
            IsSelectionEnabled="True"/>  
            </my:Chart>  
  
**The Output Will Be:**  
  
  
  
**Area Chart**  
  
**Step 1:** Now we create the Area Chart. For this first we add the following code in the .xaml file:  
  
<my:Chart  Name="AreaChart1" Title="Total Marks" Margin="33,330,0,358" HorizontalAlignment="Left" Width="379">  
                <my:AreaSeries DependentValuePath="Value"  
            IndependentValuePath="Key" ItemsSource="{Binding}"  
            IsSelectionEnabled="True"/>  
            </my:Chart>  
  
**Step 2:** After that, we add the following code in the showColumnChart() function in the .cs file :  
  
private void showColumnChart()  
{  
    List<KeyValuePair<string, int>> MyValue = new List<KeyValuePair<string, int>>();  
    MyValue.Add(new KeyValuePair<string, int>("Mahak", 300));  
    MyValue.Add(new KeyValuePair<string, int>("Pihu", 250));  
    MyValue.Add(new KeyValuePair<string, int>("Rahul", 289));  
    MyValue.Add(new KeyValuePair<string, int>("Raj", 256));  
    MyValue.Add(new KeyValuePair<string, int>("Vikas", 140));

    ColumnChart1.DataContext = MyValue;  
    **AreaChart1.DataContext = MyValue;**}  
  
**The Output Will Be:**  
  
  
  
**Line Chart**  
  
**Step 1:**Now we create the Line Chart. For this first we add the following code in the .xaml file:  
  
<my:Chart  Name="LineChart1" Title="Total Marks"  
            VerticalAlignment="Top" Margin="33,611,440,0" Height="254">  
                <my:LineSeries  DependentValuePath="Value"  
            IndependentValuePath="Key" ItemsSource="{Binding}"  
            IsSelectionEnabled="True"/>  
  
**Step 2:**After that, we add the following code in the showColumnChart() function in the .cs file :  
  
private void showColumnChart()  
{  
    List<KeyValuePair<string, int>> MyValue = new List<KeyValuePair<string, int>>();  
    MyValue.Add(new KeyValuePair<string, int>("Mahak", 300));  
    MyValue.Add(new KeyValuePair<string, int>("Pihu", 250));  
    MyValue.Add(new KeyValuePair<string, int>("Rahul", 289));  
    MyValue.Add(new KeyValuePair<string, int>("Raj", 256));  
    MyValue.Add(new KeyValuePair<string, int>("Vikas", 140));

    ColumnChart1.DataContext = MyValue;  
     AreaChart1.DataContext = MyValue;  
     LineChart1.DataContext = MyValue;

}  
  
**The Output Will Be:**  
  
**Pie Chart**  
**Step 1:** Now we create the Pie Chart. For this first we add the following code in the .xaml file:  
  
<my:Chart  Name="PieChart1" Title="Total Marks"  
            VerticalAlignment="Top" Margin="449,39,43,0" Height="262">  
                <my:PieSeries DependentValuePath="Value"  
            IndependentValuePath="Key" ItemsSource="{Binding}"  
            IsSelectionEnabled="True" />  
            </my:Chart>  
  
**Step 2:** After that, we add the following code in the showColumnChart() function in the .cs file :  
  
private void showColumnChart()  
  
{  
    List<KeyValuePair<string, int>> MyValue = new List<KeyValuePair<string, int>>();

    MyValue.Add(new KeyValuePair<string, int>("Mahak", 300));  
    MyValue.Add(new KeyValuePair<string, int>("Pihu", 250));  
    MyValue.Add(new KeyValuePair<string, int>("Rahul", 289));  
    MyValue.Add(new KeyValuePair<string, int>("Raj", 256));  
    MyValue.Add(new KeyValuePair<string, int>("Vikas", 140));

    ColumnChart1.DataContext = MyValue;  
     AreaChart1.DataContext = MyValue;  
     LineChart1.DataContext = MyValue;  
**PieChart1.DataContext = MyValue;**}  
  
  
**The Output Will be:**  
  
**Bar Chart**  
  
**Step 1:** Now we create the Bar Chart. For this first we add the following code in the .xaml file:  
  
<my:Chart  Name="BarChart1" Title="Total Marks" Margin="515,330,0,335">  
                <my:BarSeries  DependentValuePath="Value"  
            IndependentValuePath="Key" ItemsSource="{Binding}"  
            IsSelectionEnabled="True"/>  
            </my:Chart>  
  
**Step 2:** After that, we add the following code in the showColumnChart() function in the .cs file :

private void showColumnChart()  
{  
    List<KeyValuePair<string, int>> MyValue = new List<KeyValuePair<string, int>>();  
    MyValue.Add(new KeyValuePair<string, int>("Mahak", 300));  
    MyValue.Add(new KeyValuePair<string, int>("Pihu", 250));  
    MyValue.Add(new KeyValuePair<string, int>("Rahul", 289));  
    MyValue.Add(new KeyValuePair<string, int>("Raj", 256));  
    MyValue.Add(new KeyValuePair<string, int>("Vikas", 140));

    ColumnChart1.DataContext = MyValue;  
     AreaChart1.DataContext = MyValue;  
     LineChart1.DataContext = MyValue;  
    PieChart1.DataContext = MyValue;  
   **BarChart1.DataContext = MyValue;**  
}  
  
private void showColumnChart()  
{  
List<KeyValuePair<string, int>> MyValue = new List<KeyValuePair<string, int>>();  
MyValue.Add(new KeyValuePair<string, int>("Mahak", 300));  
MyValue.Add(new KeyValuePair<string, int>("Pihu", 250));  
MyValue.Add(new KeyValuePair<string, int>("Rahul", 289));  
MyValue.Add(new KeyValuePair<string, int>("Raj", 256));  
MyValue.Add(new KeyValuePair<string, int>("Vikas", 140));  
  
  
ColumnChart1.DataContext = MyValue;  
AreaChart1.DataContext = MyValue;  
LineChart1.DataContext = MyValue;  
PieChart1.DataContext = MyValue;  
BarChart1.DataContext = MyValue;  
  
}  
  
**The Output Will Be:**  
  
