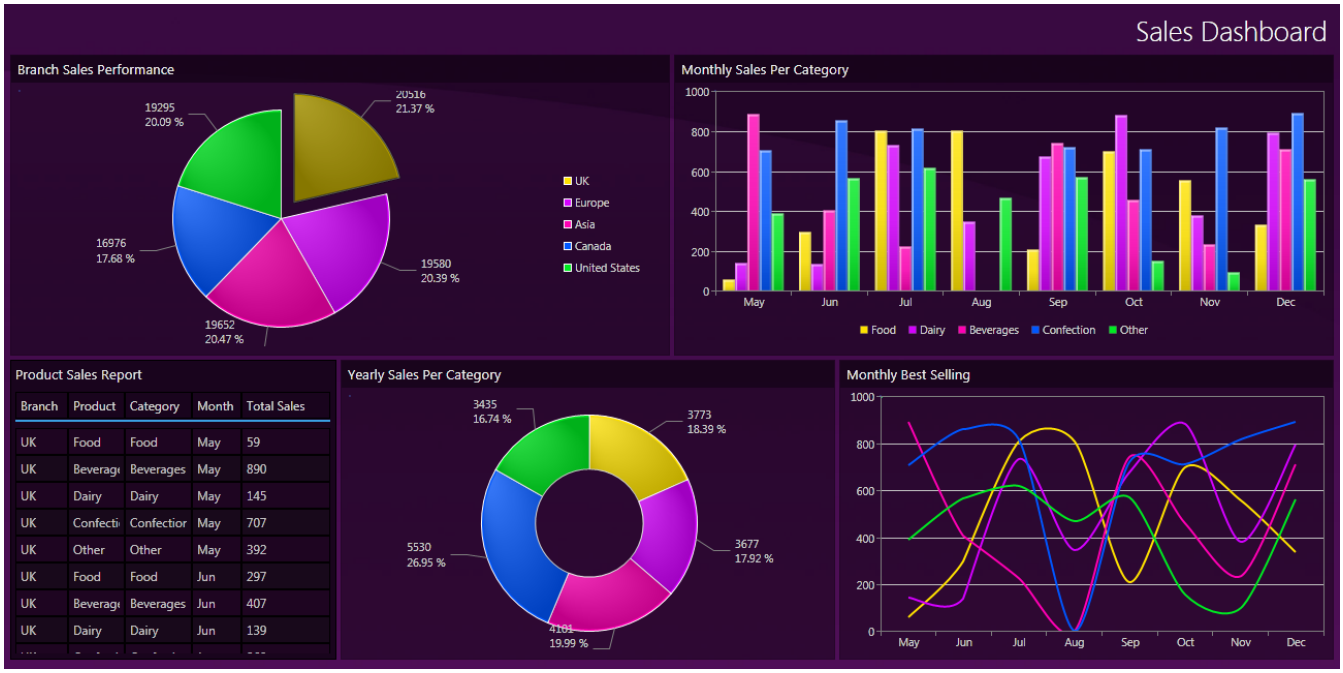
UXChart

UXChart is an advanced charting control with capability to visualize various type of data with over 20+ stunning chart types and rich user experiences to support business application development. Coupled with powerful MVVM data-binding capability,

UXChart let you easily create interactive charts with smooth animation and best-in-class user expriences that meet the most demanding requirements of today's business applications, including zooming, panning, and box selection capability. To learn more about UXChart's user experiences, see Customizing UXChart User Experiences.

On this page:



Using UXChart

In most cases, you can use UXChart to visualize your data in the following simple steps.

Binding Data

To provide data for UXChart, simply bind the **ItemsSource** property of each series to a collection that implements **INotifyCollectionChanged**, in order for the UXChart to update automatically when items are added to or removed from the source data.

There are several ways to supply data for UXChart which will be explained below.

Using Binding

The following example shows how to create a simple UXChart in XAML using binding.

XAML

```
<Intersoft:UXChart Title="Typical Use">
    <Intersoft:UXChart.Series>
        <Intersoft:ColumnSeries Title="Series Name" ItemsSource="{Binding
DataSource}"
                IndependentValuePath="IndependentValue"
DependentValuePath="DependentValue"/>
    </Intersoft:Chart.Series>
</Intersoft:Chart>
```

Binding Stacked Series

Binding data to stacked series is different than the other series. You need to bind the **ItemSource** property at **SeriesDefinition** object as follows.

XAML

```
<Intersoft:Chart x:Name="Stacked Series Sample">
    <Intersoft:Chart.Series>
        <Intersoft:StackedColumnSeries>
            <Intersoft:SeriesDefinition Title="Series 1"
                    ItemsSource="{Binding FirstCollection}"
                    DependentValuePath="{Binding DependentValue}"
                    IndependentValuePath="{Binding IndependentValue}" />
            <Intersoft:SeriesDefinition
                    Title="Series 2" ItemsSource="{Binding SecondCollection}"
                    DependentValuePath="{Binding DependentValue}"
                    IndependentValuePath="{Binding IndependentValue}" />
        </Intersoft:StackedColumnSeries>
    </Intersoft:Chart.Series>
</Intersoft:Chart>
```

The following example shows how to bind data to stacked series definition in XAML using path.

XAML

```
<Intersoft:Chart x:Name="Stacked Series Sample">
    <Intersoft:Chart.Series>
        <Intersoft:StackedColumnSeries>
            <Intersoft:SeriesDefinition Title="Series 1"
                    ItemsSource="{Binding FirstCollection}"
                    DependentValuePath="DependentValue"
                    IndependentValuePath="IndependentValue" />
            <Intersoft:SeriesDefinition Title="Series 2"
                    ItemsSource="{Binding SecondCollection}"
                    DependentValuePath="DependentValue"
                    IndependentValuePath="IndependentValue" />
        </Intersoft:StackedColumnSeries>
    </Intersoft:Chart.Series>
</Intersoft:Chart>
```

Default Axis Style

Axis is one of important part in UXChart, you can easily customize **Maximum**, **Minimum**, or **Interval** and any other specifics property of the axes (dependent and independent axis) using Style property to reflect your needs.

For most of the charts that use axis, independent axis is the horizontal one, and dependent axis is the vertical one. This setup is different in Bar series where independent axis is the vertical one, and dependent axis is horizontal one.

So the first thing to do is to specify style which is explained in following sample.

XAML

```
<Style x:Name="AxisStyle" TargetType="Intersoft:LinearAxis">
    <Setter Property="Maximum" Value="100"/>
    <Setter Property="Minimum" Value="-50"/>
    <Setter Property="Orientation" Value="X"/>
    <Setter Property="Location" Value="Left"/>
    <Setter Property="ShowGridLines" Value="True"/>
</Style>
```

Next, you can apply your style on UXChart, like the following sample

XAML

```
<Intersoft:UXChart Title="Line Chart" DependentLinearAxisStyle="{StaticResource
ResourceKey=AxisStyle}"/>
```

To learn more to styling axis using UXChart, see Working With Axis overview.

Series Binding

Defining multiple series in UXChart might introduce hassles where you might need to create separate data source in order to perform the binding. UXChart makes it easy to define multiple series by introducing series binding concept where you can bind a collection of series items into **SeriesSource**.

SeriesSource should contain a collection of object that represents each series, where each of the object should have property that represents the **ItemsSource**, **Title**, **IndependentValue** and **DependentValue**.

The following example shows a simple UXChart using series binding in XAML.

XAML

```
<Intersoft:UXChart Title="Series Binding Sample"
        SeriesSource="{Binding MedalDistributions}"
        SeriesType="{Binding SeriesBindingOption.SeriesType}"
        SeriesTitleBinding="{Binding Country}"
        SeriesItemsSourceBinding="{Binding Medals}"
        SeriesIndependentValueBinding="{Binding Host}"
        SeriesDependentValueBinding="{Binding MedalCount}"/>
```

Learn More

In the above section, you've learnt the basics of charting data visualization and learnt how to visualize your data in the simplest way. The

following links describe more advanced concepts and features of the charting control with in-depth explanation and samples.

- Understanding UXChart Series
- Working with Axis
- Working with Selection and Data Drilling
- Working with ToolTip, Nearest Data Point Finder and Data Tracker
- Customizing UXChart Elements
- Customizing UXChart User Experiences

Related Topics

- Data Visualization Overview