WPF和Silverlight版Chart3D

下面内容将详细介绍WPF和Silverlight版Chart3D。

入门

查看下面内容开始了解和学习WPF和Silverlight版Chart3D。

WPF和Silverlight版帮助

关于安装WPF版ComponentOne Studio 许可，信息，技术支持，命名空间以及通过控件创建一个工程，请访问WPF版本入门。

关于安装Silverlight版ComponentOne Studio 许可，信息，技术支持，命名空间以及通过控件创建一个工程，请访问Silverlight版本入门。

主要功能

WPF和Silverlight版 Chart3D提供以下独有功能：

- **曲面图图表类型**
  WPF和Silverlight 版Chart3D包括6种不同的曲面图图表类型。当创建曲面图类型的图表时，您可以在线框图，轮廓图以及分区图之间进行选择。自定义每一个轴的轮廓和网格是否出现。更多信息请参见图表类型。
- **显示图例**
  为分区图显示图表图例。更多信息，请参见添加一个图表图例。
- **设置旋转角度和仰角**
  通过设置Azimuth 以及Elevation 属性旋转图表至任意角度。更多信息，请参见旋转和改变图表仰角。
- **Floors 和Ceilings**
  由图表决定的轮廓以及分区可以投影显示在绘图立方体区域的顶部或者底部。在图表的顶部和底部的显示轮廓和区域，除了三维模型显示以外，以提供一个二维表示。相关示例，请参见添加Floors和Ceilings。
- **显示二维投影**
  轻松创建平面（或二维）三维图表，如热地图。更多信息，请参见创建一个二维图表。
- **创建自定义坐标轴注释**
  通过使用AnnoTemplate属性自定义坐标轴标签的精确的布局以及值。更多信息，请参见查看自定义坐标轴的注释模板。

快速入门：WPF和Silverlight版Chart3D

第一步：将Chart3D添加到您的工程

在这一步，您将开始在Visual Studio中创建一个使用WPF和Silverlight版Chart3D 图表的应用程序。

1. 在Visual Studio中创建一个新的WPF或Silverlight应用程序。
2. 双击位于工具栏的C1Chart3D控件（位于在线文档'C1Chart3D Class'章节），以将其添加至您的窗体。XAML标记将类似于下面的代码片段所示：

XAML - WPF

```
<Window x:Class="WFPChart3DQuickstart.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
      xmlns:c1chart3d="http://schemas.componentone.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525" >
    <Grid>
        <c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left"
Margin="10,10,0,0" Name="c1Chart3D1" VerticalAlignment="Top" Width="200">
            <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
```

```
        </c1chart3d:C1Chart3D>
    </Grid>
</Window>
```

XAML - Silverlight

```
<UserControl xmlns:c1chart3d="http://schemas.componentone.com/winfx/2006/xaml"
    x:Class="MySilverlightApplication.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
        <c1chart3d:C1Chart3D Name="c1Chart3D1"/>
    </Grid>
</UserControl>
```

在下一步中，您将添加自己的数据到图表中。

第二步：添加数据

WPF和Silverlight版Chart3D 支持数据值定义为一个二维数组的z值，其在第一个索引对应的X坐标，第二个索引值对应Y坐标。Chart3D 最常见的用途之一就是绘制3D功能。让我们绘制一个函数定义如下：
z(x,y) = x*x - y*y;

位于范围
-1 <= x <= 1

-1 <= y <= 1

要做到这一点，请遵循以下步骤：

1. 在你的工程中选择视图|代码。
2. 在页面顶部添加如下语句：

C# - WPF

```
using C1.WPF.C1Chart3D;
```

C# - Silverlight

```
using C1.Silverlight.Chart3D
```

3. 添加以下代码用于定义数据：

C#

```
public MainWindow()
        {
            InitializeComponent();
            c1Chart3D1.Children.Clear();

            // 创建 2D 数组  10x10
            int xlen = 10, ylen = 10;
```

```
            var zdata = new double[xlen, ylen];
            double stepx = 2.0 / (xlen - 1);
            double stepy = 2.0 / (ylen - 1);
            // 计算范围内的全部点
            for (int ix = 0; ix < xlen; ix++)
                for (int iy = 0; iy < ylen; iy++)
                {
                    double x = -1.0 + ix * stepx; //  -1 <= x <= 1
                    double y = -1.0 + iy * stepy; //  -1 <= x <= 1
                    zdata[ix, iy] = x * x - y * y;
                }
            // 创建数据系列
            var ds = new GridDataSeries();
            ds.Start = new Point(-1, -1); // 起始点x,y
            ds.Step = new Point(stepx, stepy); // 阶梯x,y
            ds.ZData = zdata; // z-values
            // 添加系列至图表
            c1Chart3D1.Children.Add(ds);
        }
```

在下一步中，您将改变图表的外观。

第三步：改变图表外观

在这一步中您将图表类型更改为SurfaceZone，设置图表底部的外观，旋转该图表，并改变其仰角。

1. 在您的窗体中选中C1Chart3D （在线文档中位于'C1Chart3D Class' ），并单击查看|属性窗体。
2. 通过点击ChartType（位于在线文档'ChartType 属性'）属性旁边的下拉箭头以更改图表类型，并选择SurfaceZone。
3. 单击FloorAppearance （位于在线文档'FloorAppearance属性'）属性，并选择ZoneContour。
4. 在Elevation（位于在线文档'Elevation 属性'）属性一栏，填写170。
5. 设置Azimuth（位于在线文档'Azimuth 属性'）属性的值为20。
   XAML标记将类似于下面的代码所示：

XAML - WPF

```
<Window x:Class="WFPChart3DQuickstart.MainWindow"
        xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
        xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
        Title="MainWindow" Height="350" Width="525"
xmlns:c1chart3d="http://schemas.componentone.com/xaml/c1chart">
    <Grid>
        <c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left"
Margin="10,10,0,0" Name="c1Chart3D1" VerticalAlignment="Top" Width="200"
ChartType="SurfaceZone" CeilAppearance="None" FloorAppearance="ZoneContour"
Elevation="170" Azimuth="20">
            <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
        </c1chart3d:C1Chart3D>
    </Grid>
</Window>
```

XAML - Silverlight

```
<UserControl xmlns:c1chart3d="clr-
namespace:C1.Silverlight.Chart3D;assembly=C1.Silverlight.Chart3D"
x:Class="SilverlightApplication12.MainPage"
    xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
```

```
    xmlns:d="http://schemas.microsoft.com/expression/blend/2008"
    xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
    mc:Ignorable="d"
    d:DesignHeight="300" d:DesignWidth="400">
    <Grid x:Name="LayoutRoot" Background="White">
<c1chart3d:C1Chart3D Name="c1Chart3D1" ChartType="SurfaceZone"
FloorAppearance="ZoneContour" Elevation="170" Azimuth="20"/>
<c1chart3d:C1Chart3D />
    </Grid>
</UserControl>
```

在下一步中，您将为图表添加一个图例。

第四步：添加一个图例

在这一步中，您将为图表添加一个图例，仅需通过设置一个属性。

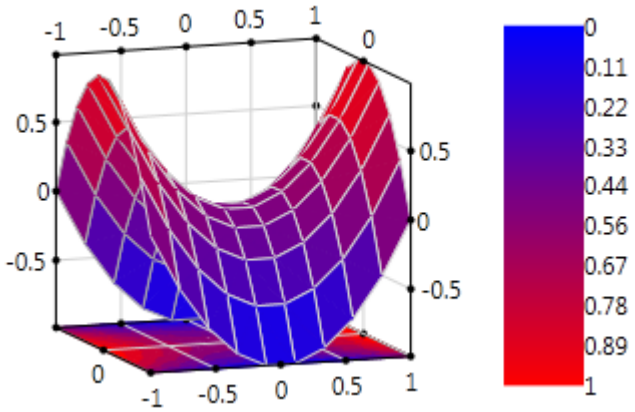
1. 在您的窗体中选中C1Chart3D控件（在线文档'C1Chart3D类'），单击查看|属性窗体。
2. 通过单击Legend属性旁边的下拉箭头添加一个图例，并选择C1Chart3DLegend （在线文档'C1Chart3DLegend类'）。

在下一步中，您将运行该项目并查看新创建的图表。

第五步：运行该工程

在这一步中，您将运行该项目以查看图表。

选择开始|调试或按F5键。您会看到Chart3D图例。



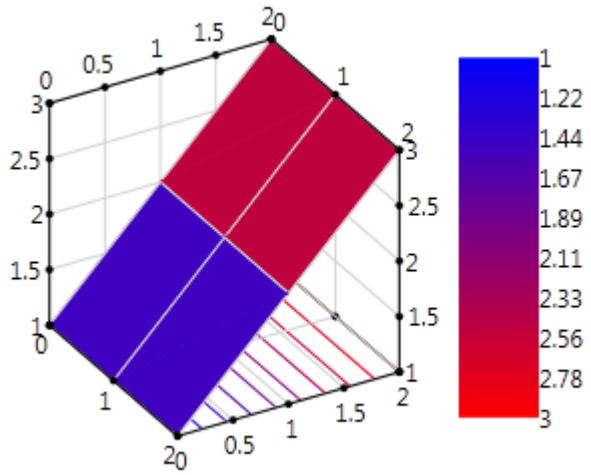
恭喜完成！您已经完成了WPF和Silverlight版Chart3D的快速入门。

XAML快速参考

示例：设置一个三维曲面图

下面的代码显示了如何声明C1Chart3D 控件，设置其ChartType，FloorAppearance，GridDataSeries，并添加一个C1Chart3DLegend 控件。 XAML可以添加在<Grid>...</Grid>标签内部。

下面的图表是通过以下XAML代码产生：



XAML

```
<Grid>
    <c1chart3d:C1Chart3D Name="c1Chart3D1" ChartType="SurfaceZone"
FloorAppearance="Contour">
        <c1chart3d:C1Chart3D.Legend>
            <c1chart3d:C1Chart3DLegend />
        </c1chart3d:C1Chart3D.Legend>
        <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
    </c1chart3d:C1Chart3D>
</Grid>
```

使用WPF和Silverlight版 Chart3D

以下内容介绍了如何使用WPF和Silverlight版Chart3D。

GridDataSeries数据布局

WPF 和Silverlight 版Chart3D中的数据在GridDataSeries类（位于在线文档'GridDataSeries 类'）中进行定义。它提供了与数据相关的下列属性：

属性	描述
ZData	获取或设置Grid上的二维数组。
Start	获取或设置数据的起始点。
Step	获取或设置Grid数据的步进值。
ContourData	获取或设置轮廓数据的二维数组（四维图）。

ZData（在线文档'ZData属性'）/ContourData（在线文档'CountourData属性'）属性的二维数组的第一个索引关联到X，第二个索引关联到Y。ZData的布局如下表所示。

	Start.X	Start.X + Step.X	Start.X + 2*Step.X
Start.Y	ZData[0,0]	ZData[1,0]	ZData[2,0]
Start.Y + Step.Y	ZData[0,1]	ZData[1,1]	ZData[2,1]

Start.Y + 2*Step.Y	ZData[0,2]	ZData[1,2]	ZData[2,2]

数组中的元素ZData[0,0]关联到点 (Start.X, Start.Y)，ZData[1,0]是位于(Start.X+Step.X, Start.Y)的 z-value，依此类推。

图表类型

使用内置类型是设置图表外观最简单的方法。例如，为了设置一个线框曲面图，为ChartType属性（在线文档'ChartType属性'）指定相应的字符串：

XAML

```
<C1_WPF_C1Chart3D:C1Chart3D ChartType="SurfaceWireframe"/>
```

可用的图表类型由Chart3DType枚举（在线文档'Chart3DType枚举'）的成员指定。

下表展示了全部可用的图表类型的列表。

名称	描述
SurfaceWireframe	线框曲面图
SurfaceWireframeContour	轮廓线框线图
Surface	曲面图
SurfaceContour	轮廓曲面图
SurfaceZone	分区曲面图
SurfaceZoneContour	等高线图面图

曲面图检查数据的分布，并绘制轮廓线界定每一个轮廓等级。您可以使用ContourLevelsCount 属性（在线文档'ContourLevelsCount 属性'）设置轮廓线的等级。

WPF和Silverlight版 Chart3D还支持将网格线添加到X轴，Y轴或全部两个坐标轴。使用SurfaceMeshAppearance 属性（在线文档'SurfaceMeshAppearance 属性'）以决定是否显示网格线。

添加一个图表图例

WPF和Silverlight版 Chart3D允许你显示轮廓或分区图的图例。图里元素显示图表每一个数据系列的信息。图表中的图例显示了物理颜色和数据系列之间的映射关系。这是由C1Chart3D（在线文档'C1Chart3D类'）的Legend属性进行控制。

为添加一个图例至图表，请遵循以下步骤：

1. 向窗体添加一个C1Chart3D控件。
2. 设置ChartType（在线文档'ChartType 属性'）属性的值为SurfaceZone 或者SurfaceZoneContour。XAML将类似于下面的代码所示：

   XAML

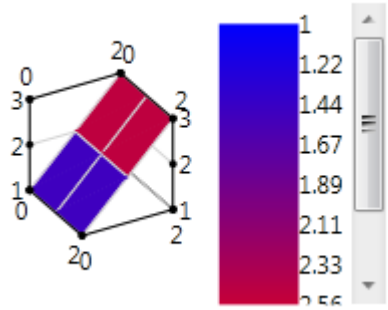
   ```
   <c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left"
   Margin="137,77,0,0" Name="c1Chart3D1" VerticalAlignment="Top" Width="200"
   ChartType="SurfaceZone">
             <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
   </c1chart3d:C1Chart3D>
   ```

3. 设置Legend 属性（在线文档'Legend属性'）的值为C1Chart3DLegend（在线文档'C1Chart3DLegend类'）。现在，XAML代码看起来像这样：

XAML

```
<c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left"
Margin="178,85,0,0" Name="c1Chart3D1" VerticalAlignment="Top" Width="200"
ChartType="SurfaceZone">
            <c1chart3d:C1Chart3D.Legend>
                <c1chart3d:C1Chart3DLegend />
            </c1chart3d:C1Chart3D.Legend>
            <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
</c1chart3d:C1Chart3D>
```

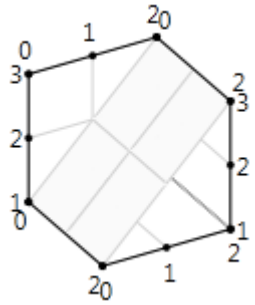
您的图表将看起来如下图所示：



请注意，图例仅在轮廓或分区图中可见。

旋转和升降图表

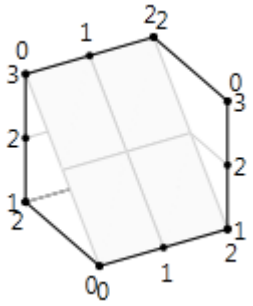
可以很容易地通过Azimuth属性（在线文档'Azimuth属性'）旋转图表。C1Chart3D （在线文档'C1Chart3D 类'）的默认旋转角度是30度。



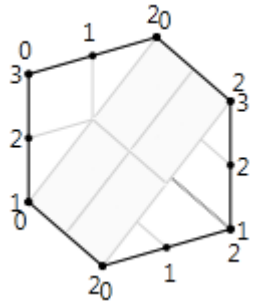
下面的XAML设置Azimuth 属性的值为120度。

XAML

```
<c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left" Margin="96,51,0,0"
Name="c1Chart3D1" VerticalAlignment="Top" Width="200" Azimuth="120">
```



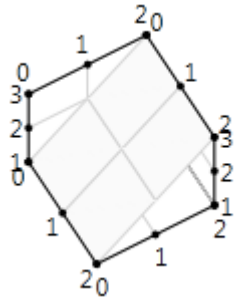
每一个图表具有一个仰角，或者在X轴上方的倾斜角。当您第一次向窗体添加一个图表时，其仰角默认为150度。



为改变该倾斜角度，可以使用Elevation（在线文档'Elevation属性'）属性。下面的XAML设置Elevation属性的值为120。

XAML

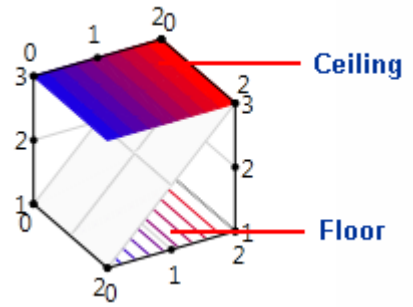
```
<c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left" Margin="96,51,0,0"
Name="c1Chart3D1" VerticalAlignment="Top" Width="200" Azimuth="30" Elevation="120">
```



您还可以使用Elevation 属性创建一个二维图表。更多信息，请参见创建一个二维图表。

添加Floor和Ceiling

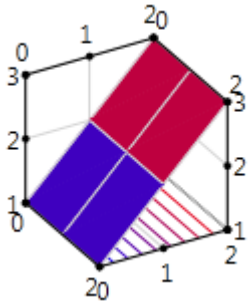
WPF和Silverlight版 Chart3D允许您通过在绘图区域立方体的顶部，Ceiling区域，底部以及Floor区域显示轮廓或分区以允许图表以二维方式显示。



为确定Floor的外观，设置FloorAppearance 属性（在线文档'FloorAppearance 属性'）。在以下示例中，该XAML片段设置一个SurfaceZone 图表的FloorAppearance 属性的值为Contour。

XAML

```
<c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left" Margin="234,132,0,0"
Name="c1Chart3D1" VerticalAlignment="Top" Width="200" ChartType="SurfaceZone"
FloorAppearance="Contour">
        <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
</c1chart3d:C1Chart3D>
```

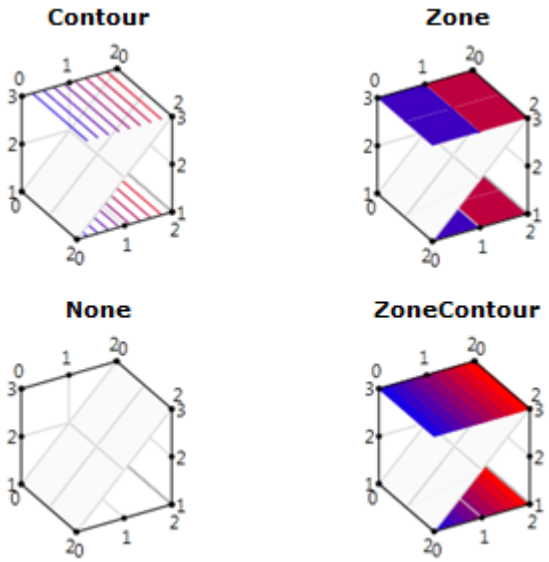


为确定Ceiling的外观，设置CeilAppearance 属性（在线文档'CeilAppearance 属性'）。在下面的例子中，该XAML片段设置一个SurfaceContour 图表的CeilAppearance 属性的值为Zone。

XAML

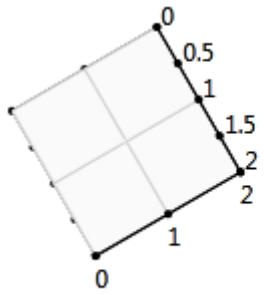
```
<c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left" Margin="234,132,0,0"
Name="c1Chart3D1" VerticalAlignment="Top" Width="200" ChartType="SurfaceContour"
CeilAppearance="Zone">
    <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
</c1chart3d:C1Chart3D>
```

FloorAppearance 以及CeilAppearance 属性提供四外观选项：

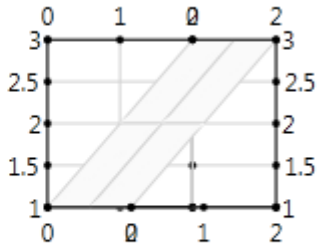


创建一个二维图表

通过修改图表的Elevation属性（在线文档'Elevation属性'），您可以容易地创建一个平面的三围图表，比如说热点地图。每个图表有一个仰角，或一个X轴上访的倾斜角。当您第一次向窗体添加一个图表时，默认情况下，仰角为150度。您可以通过设置Elevation 属性为90或180将图表平面化，取决于您希望如何将其相对于X轴平面化。



Elevation = "90"



Elevation = "180"

下面的XAML片段设置一个图表的Elevation属性的值为90。

XAML

```
<c1chart3d:C1Chart3D Height="150" HorizontalAlignment="Left" Margin="234,139,0,0"
Name="c1Chart3D1" VerticalAlignment="Top" Width="200" Elevation="90">
     <c1chart3d:GridDataSeries ZDataString="1 1 1,2 2 2,3 3 3" />
</c1chart3d:C1Chart3D>
```

创建自定义坐标轴注释

三维图表具有X，Y和Z轴，这些坐标轴可以由C1Chart3D对象控制。您可以通过C1Chart3D 对象自定义坐标轴的外观，然后通过AxisX，AxisY，以及AxisZ属性设置 X，Y和Z轴的新外观。

本章节描述了最常见的坐标轴设置 场景，用于将图表变得更加可读，比如说加标签，缩放，以及格式化。

自定义坐标轴标注模板

以下示例演示如何使用AnnoTemplate 属性（在线文档'AnnoTemplate 属性'）创建一个颜色取决于相应的坐标轴的值的坐标轴标签。

添加以下XAML代码至您的 MainWindow.xaml 或者 MainPage.xaml文件：

> **注意：**在一个Silverlight应用程序，您应当添加<Page.Resources> </Page.Resources>标签以取代<Window.Resources>标签。

XAML

```
<Window.Resources>
    <!-- 转换器实例 -->
    <local:LabelToColorConvereter x:Key="cnv" />
  </Window.Resources>
    <Grid>
    <c1chart3d:C1Chart3D Name="c1Chart3D1" >
      <c1chart3d:GridDataSeries ZDataString="-1 -1 -1,0 0 0, 1 1 1" />
      <c1chart3d:C1Chart3D.AxisZ>
        <c1chart3d:Axis3D>
          <!-- 设置坐标轴注释模板 -->
          <c1chart3d:Axis3D.AnnoTemplate>
            <DataTemplate >
              <!-- DataContext 是坐标轴标签（字符串），通过转换器设置颜色-->
                <TextBlock Width="20" Height="12" Text="{Binding}"
Foreground="{Binding Converter={StaticResource cnv}}" />
            </DataTemplate>
          </c1chart3d:Axis3D.AnnoTemplate>
        </c1chart3d:Axis3D>
```

```
    </c1chart3d:C1Chart3D.AxisZ>
  </c1chart3d:C1Chart3D>
</Grid>
```

选择视图|代码并添加以下代码。绑定转换器按照不同的值选择不同的坐标轴标签的画刷。

C#

```
public class LabelToColorConvereter : IValueConverter
{
  static Brush belowZero = new SolidColorBrush(Colors.Blue);
  static Brush aboveZero = new SolidColorBrush(Colors.Red);
  static Brush zero = new SolidColorBrush(Colors.DarkGray);
  public object Convert(object value, Type targetType, object parameter,
    System.Globalization.CultureInfo culture)
  {
    // 表示坐标轴标签的字符串
    string s = value as string;
    if (!string.IsNullOrEmpty(s))
    {
      var dv = double.Parse(s);
      // 返回由值决定的画刷
      if (dv < 0)
        return belowZero;
      else if (dv > 0)
        return aboveZero;
      else
        return zero;
    }
    return value;
  }

  public object ConvertBack(object value, Type targetType, object parameter,
    System.Globalization.CultureInfo culture)
  {
    throw new NotImplementedException();
  }
}
```

坐标轴标题

向坐标轴添加一个标题以声明该坐标轴表示的图表值。带有指定字体的标题可以添加到任何坐标轴。

添加一个坐标轴标题：

利用坐标轴Title（在线文档'Title属性'）属性向坐标轴添加一个标题。

1. 在Visual Studio属性窗口，展开AxisX，AxisY，或者AxisZ属性。
2. 在Title属性一栏键入一个标题。
3. 为删除标题，请删除该段文本。

DirectX渲染

注意：本主题仅适用于WPF应用程序。

DirectX渲染模式允许您创建一个高性能的曲面图表，通过Direct3D渲染，可以支持更大的数据集合。为了实现DirectX渲染模式，仅需设置RenderMode（在线文档'RenderMode属性'）属性的值为Direct3D。

为了在您的WPF应用程序中使用Direct3D渲染模式，您还应当将以下程序集和您的应用程序部署在一起。
在开发过程中，这些组件应当可以在 C1.WPF.C1Chart3D 所在的同一目录中找到，但是不需要再您的工程中直接引用。

- C1.WPF.DX.4.dll

Direct3D的渲染的系统要求包括：

- DirectX 10或更高版本
- D3D9Overlay支持

Direct3D模式在一些虚拟机（Hyper-V）中，或者通过远程桌面访问时不起作用，因为所需的硬件层不提供支持。

API引用

下面内容详细介绍了WPF版Chart3D的API。