

# Sparkline for WPF

2018.04.11 更新

グレースィティ株式会社

## 目次

<a href="#">Sparkline for WPF の概要</a>	2
<a href="#">WPF Edition のヘルプ</a>	2
<a href="#">主要な機能</a>	2
<a href="#">クイックスタート: Sparkline for WPF</a>	2-4
<a href="#">機能</a>	4
<a href="#">軸</a>	4
<a href="#">日付軸</a>	4-5
<a href="#">データ連結</a>	5
<a href="#">スパークラインのタイプ</a>	5-7
<a href="#">スパークラインタイプの設定</a>	7
<a href="#">マーカー</a>	7-9
<a href="#">マーカーの色のカスタマイズ</a>	9-10
<a href="#">タスク別ヘルプ</a>	10
<a href="#">データ管理コントロールでのスパークラインの使用</a>	10-12

## Sparkline for WPF の概要

**ComponentOne** は、データの傾向や変動を表現するための軽量のデータ視覚化コントロールとして **Sparkline for WPF** を導入しています。スパークラインはインライングラフとして設計されており、極めて簡約された形式でデータをプロットします。標準のチャートとは異なり、スパークラインは、座標、軸、凡例、タイトルなどの基本的なチャート要素なしで描画されます。

スパークラインチャートのさらに興味深い点は、そのサイズです。スパークラインチャートはコンパクトサイズなので、グリッドやダッシュボードに簡単に埋め込むことができます。

## WPF Edition のヘルプ

**ComponentOne WPF Edition** のインストール、ライセンス、テクニカルサポート、名前空間、およびこれらのコントロールを含むプロジェクトの作成方法については、「[WPF Edition の開始](#)」を参照してください。

## 主要な機能

軽量のデータ視覚化コントロールであるスパークラインは、傾向線や変動曲線などを描画するために適したさまざまな機能を提供しています。ここでは、スパークラインの主な機能について簡単に説明します。

主要な機能	説明
スパークラインのタイプ	Sparkline コントロールは、さまざまなビジネスニーズに対応する縦棒グラフ、折れ線グラフ、勝敗チャートの 3 つのタイプをサポートします。
マーカー	データ値を強調表示するために、スパークラインは、ユーザーの要件に合わせてカスタマイズできるマーカーをサポートします。
データ連結	Sparkline コントロールは、データ値の列挙可能コレクションに簡単に連結できます。
軸	他のチャートと同様に、スパークラインは X 軸をサポートします。これは、ユーザーの要件に合わせて表示/非表示にできます。デフォルトでは、スパークラインの X 軸は非表示のままです。X 軸を使用して一定期間のデータを表示することもできます。詳細については、「 <a href="#">日付軸</a> 」を参照してください。
外観	<b>C1Sparkline</b> クラスには、コントロールの外観をカスタマイズするための多数のプロパティがあります。マーカーを設定するほかに、軸、データポイント(最初と最後)、系列全体の色もカスタマイズできます。

## クイックスタート: Sparkline for WPF

このクイックスタートセクションは、Sparkline コントロールを理解することを目的としています。最初に Visual Studio で WPF アプリケーションを作成し、Sparkline コントロールを追加し、サンプルデータを挿入します。

Sparkline コントロールが実行時にどのように表示されるかを確認するには、次の手順を実行します。

- **手順 1: アプリケーションの設定**
- **手順 2: 実行時にコントロールに設定するサンプルデータの追加**
- **手順 3: コントロールをサンプルデータに連結するコードの追加**

カスタマイズなしの基本的な Sparkline コントロールは、次のように表示されます。



## 手順 1: アプリケーションの設定

1. Visual Studio で WPF アプリケーションを作成します。
2. C1Sparkline コントロールを MainWindow にドラッグアンドドロップします。
3. XAML ビューを編集して、関連する名前空間とコントロールの基本的なプロパティを設定します。

```
XAML copyCode  
  
<Window x:Class="Sparkline_QuickStart.MainWindow"  
  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"  
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"  
  xmlns:cl="http://schemas.componentone.com/winfx/2006/xaml"  
  Title="MainWindow" Height="350" Width="525">  
  <Grid>  
    <cl:C1Sparkline x:Name="sparkline" Height="250" Width="250"  
      SparklineType="Column"/>  
  </Grid>  
</Window>
```

## 手順 2: 実行時にコントロールに設定するサンプルデータの追加

この手順では、スパークラインチャートにプロットする数値データポイントの列挙可能なコレクションを返すクラスを追加します。このコード例では、数値のコレクションを返すクラス **SampleData.cs** を追加します

1. ソリューションエクスプローラーで、プロジェクト名を右クリックし、**[追加]**→**[クラス]**を選択します。
2. クラスの名前(ここでは SampleData)を指定し、**[追加]**をクリックします。
3. 1つのメソッドで、数値データの列挙可能なコレクションを作成し、それを返すコードを追加します。

## 手順 3: コントロールをサンプルデータに連結するコードの追加

上記の手順で追加された **SampleData.cs** クラスは、数値のコレクションを返します。この手順では、実行時にデータがプロットされるように、このコレクションを Sparkline コントロールに連結します。それには、**SampleData** クラスのオブジェクトを作成し、それを Sparkline コントロールの **Data** プロパティに割り当てます。

1. MainPage.xaml.cs クラスファイルに切り替え、次の Import 文を追加します。
2. 次のように相互作用ロジックを編集して、SampleData クラスのオブジェクトを作成し、それをコントロールの **Data** プロパティに割り当てます。
3. **[F5]**キーを押してアプリケーションを実行し、Sparkline コントロールが実行時にどのように表示されるかを確認しま

す。

## 学習内容

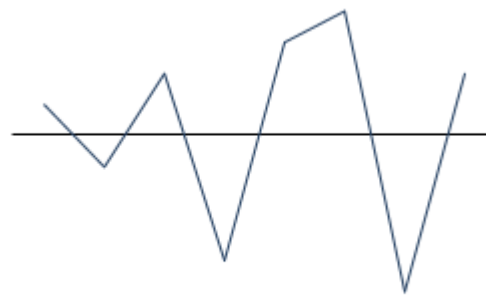
おめでとうございます。これで、WPF アプリケーションを作成し、Sparkline コントロールを追加してサンプルデータに連結し、データポイントの変化を表示できました。Sparkline は、軸、凡例、タイトルなどの基本的なチャート要素なしでデータの変化を示すチャートをレンダリングします。

## 機能

### 軸

デフォルトでは、Sparkline コントロールは軸をレンダリングしません。ただし、スパークラインは X 軸をサポートします。これは、ユーザーの要件に合わせて表示/非表示にできます。デフォルトでは、スパークラインの X 軸は非表示のままです。

軸を表示するには、**DisplayXAxis** プロパティを true に設定する必要があります。Sparkline コントロールの X 軸は、次のように表示されます。



x軸を表示するスパークライン

### XAML の場合

XAML

copyCode

```
<cl:C1Sparkline x:Name="sparkline" Height="250" Width="250" DisplayXAxis="True" />
```

### コードの場合

## 日付軸

スパークラインは日付軸をサポートし、いくつかの日付にわたるデータを X 軸にプロットします。日付軸は、不定な間隔でデータを表示できる形態の X 軸に過ぎません。日付軸は日付のコレクションを受け取り、これらの日付を X 座標としてデータをプロットします。この方法で、いくつかの日付にわたるデータをプロットできます。

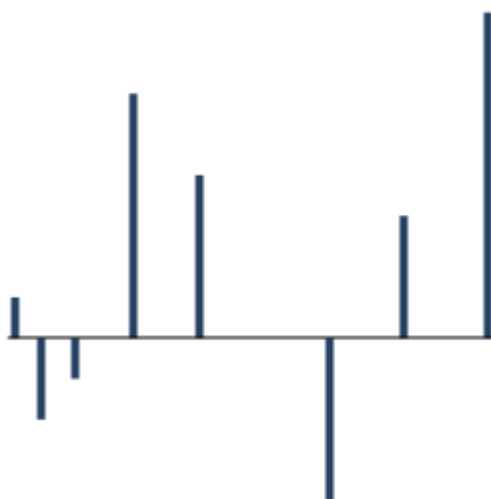
たとえば、次のような毎月の売上利益の数字からその傾向を表示したいとします。次の表で、負の値は損失を表します。

日付	売上利益(10 万単位)
2016-01-01	1.0
2016-01-15	-2.0
2016-02-01	-1.0
2016-03-16	6.0

# Sparkline for WPF

2016-04-06	4.0
2016-06-11	-4.0
2016-07-19	3.0
2016-09-01	8.0

**C1Sparkline** クラスには **DisplayDateAxis** プロパティがあり、これを true に設定すると、いくつかの日付にわたるデータを表示できます。次の図は、Sparkline コントロールに不定な間隔でプロットされたデータです。



日付軸を設定し、不定な間隔でデータをプロットするには、次の手順を実行します。

日付軸を設定するには、X 軸となる日付の列挙可能なコレクションを提供し、これを **DateAxisData** プロパティに設定します。この例では、「[クイックスタート](#)」で作成したサンプルを使用します。


1. クイックスタートで作成した SampleData.cs クラスに次のコードを追加します。
2. MainWindow.xaml.cs クラスに切り替え、相互作用ロジックを編集して、スパークラインタイプを設定し、日付コレクションを **DateAxisData** プロパティに割り当てます。

## データ連結

データは、データ視覚コントロールの中核です。Sparkline コントロールは、数値データ値の列挙可能なコレクションに連結できます。スパークラインは日付軸をサポートするため、X 軸となる日付のコレクションにも連結できます。

**C1Sparkline** クラスには、Sparkline コントロールとデータを連結する 2 つのプロパティがあります。

プロパティ	説明
<b>Data</b>	このプロパティは、Sparkline コントロールを数値のコレクションに連結します。
<b>DateAxisData</b>	このプロパティは、Sparkline コントロールを日付のコレクションに連結します。

 これらのプロパティを設定してデータ連結を行う方法については、「[クイックスタート](#)」および「[日付軸](#)」を参照してください。

## スパークラインのタイプ


スパークラインは、さまざまなコンテキストでデータを視覚化するために、折れ線、縦棒、勝敗の 3 種類のチャートタイプをサポートします。たとえば、折れ線グラフは、連続したデータの視覚化に役立ち、縦棒グラフのスパークラインは、データの比較を目的とする場合に使用されます。同様に、勝敗のスパークラインは、真偽(勝敗)の状態を視覚化するために最適です。

## 折れ線スパークライン

折れ線スパークラインは、データの傾向を視覚化する必要がある場合に最適な直線グラフとして描画されます。このタイプのスパークラインは、売上高や株価の視覚化によく使用されます。

Sparkline コントロールの **SparklineType** プロパティを Line に設定すると、次のように表示されます。



 デフォルトでは、スパークラインは折れ線スパークラインとしてレンダリングされます。

## 縦棒スパークライン

縦棒スパークラインは、データポイントを縦棒でレンダリングします。このタイプのスパークラインは、データの比較を視覚化するために適しています。縦棒スパークラインでは、正のデータポイントは上向きに、負のデータポイントは下向きに描画されます。

Sparkline コントロールの **SparklineType** プロパティを Column に設定すると、次のように表示されます。



## 勝敗スパークライン

勝敗スパークラインは、データポイントをサイズが同じ縦棒で上下方向に表示します。このタイプのスパークラインは、勝敗(損得)の状態を視覚化するために使用されます。上方向に描画された縦棒は勝ちを示し、下方向の縦棒は負けを示します。

Sparkline コントロールの **SparklineType** プロパティを Win-Loss に設定すると、次のように表示されます。



## スパークラインタイプの設定

**C1Sparkline** クラスには、XAML またはコードでチャートタイプを設定するための **SparklineType** プロパティがあります。**SparklineType** プロパティは、**SparklineType** 列挙に含まれる次の値を受け取ります。

- **Column** - 縦棒スパークラインを指定します。
- **Line** - 折れ線スパークラインを指定します。
- **Winloss** - 勝敗スパークラインを指定します。

### XAML の場合

XAML でスパークラインのチャートタイプを設定するには、**SparklineType** プロパティを特定のチャートタイプに設定する必要があります。

XAML

copyCode

```
<c1:C1Sparkline x:Name="sparkline" Height="150" Width="250" SparklineType="Column" />
```

### コードの場合

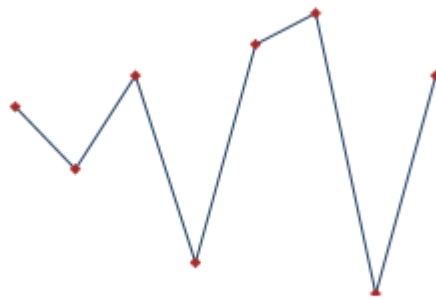
**SparklineType** プロパティを使用して、コードでスパークラインのチャートタイプを設定することもできます。

## マーカー

Sparkline コントロールは、一連のデータポイントをプロットすることで、傾向や変動を表現することを主な目的としています。データは、スパークラインタイプに応じて、折れ線グラフ、勝敗傾向、または縦棒として、チャート固有の要素やカスタマイズなしで描画されます。ただし、データポイントの表示方法をカスタマイズしたい場合もあります。たとえば、スパークラインのデータポイントを指示したり、強調表示したい場合です。そのために、スパークラインには、表示/非表示を明示的に設定してデータポイントを強調表示できるマーカーが付属しています。

**C1Sparkline** クラスには **ShowMarkers** プロパティがあり、これを true に設定すると、すべてのデータポイントが強調表示されます。Sparkline コントロールのすべてのデータポイントをマーカーで強調表示すると、次のようになります。




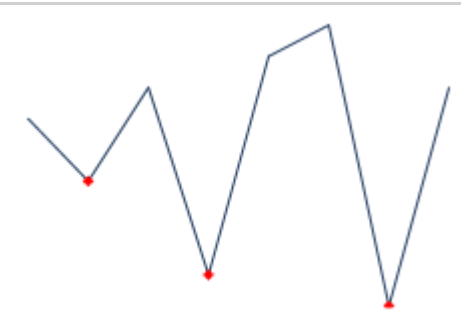



データポイントをマーカーで強調表示するスパークライン

**ShowMarkers** プロパティは、スパークラインのすべてのデータポイントを強調表示します。ただし、ビジネスニーズによっては、特定の値を強調表示したい場合もあります。そのために、Sparkline コントロールでは、特定のデータポイントにもマーカーを適用できます。次のプロパティを同様にコードで設定できます。

プロパティ	説明	出力
<b>ShowFirst</b>	このプロパティをコード内で true に設定して、右の画像のように、出力の最初のデータポイントを強調表示できます。デフォルトでは、最初のデータポイントがえび茶色のマーカーで強調表示されます。	
<b>ShowLast</b>	このプロパティをコード内で true に設定して、右の画像のように、出力の最後のデータポイントを強調表示できます。デフォルトでは、最後のデータポイントが緑色のマーカーで強調表示されます。	
<b>ShowHigh</b>	このプロパティをコード内で true に設定して、右の画像のように、出力の最大のデータポイントを強調表示できます。デフォルトでは、最大のデータポイントが赤色のマーカーで強調表示されます。	

# Sparkline for WPF

<b>ShowLow</b>	このプロパティをコード内で true に設定して、右の画像のように、出力の最小のデータポイントを強調表示できます。デフォルトでは、最小のデータポイントが青色のマーカで強調表示されます。	
<b>ShowNegative</b>	このプロパティをコード内で true に設定して、右の画像のように、出力のすべての負のデータポイントを強調表示できます。デフォルトでは、負のデータポイントが赤色のマーカで強調表示されます。	

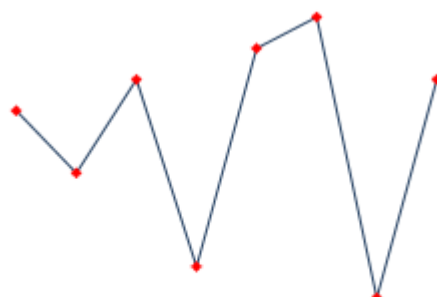
 縦棒と勝敗のスパークラインでは、マーカを使用してデータポイントを強調表示できません。

## マーカの色のカスタマイズ

マーカは、各データポイントをデフォルトの色で指示または強調表示します。たとえば、**ShowNegative** プロパティを true に設定すると、負のデータポイントが赤色で強調表示され、**ShowLast** プロパティを true に設定すると、最後のデータポイントが緑色で強調表示されます。ただし、このデフォルトの動作を変更して、選択した色でデータポイントを強調表示することで、スパークラインの外観をさらにカスタマイズできます。

**C1Sparkline** クラスには、コードでマーカの色をカスタマイズできるさまざまなプロパティがあります。たとえば、**MarkersColor** プロパティは、各データポイントのマーカの色を指定します。

**MarkersColor** プロパティを Red に設定すると、Sparkline コントロールは次のように表示されます。



データポイントを赤色で強調表示するマーカ

プログラムから別の色でデータマーカを強調表示するには、コードで **MarkersColor** プロパティを設定します。

以下は、特定のデータポイントを強調表示するためのマーカ色をカスタマイズするために使用できるプロパティのリストです。

プロパティ	説明
<b>FirstMarkerColor</b>	このプロパティを設定すると、スパークラインの最初のデータポイントのマーカの色を指定できます。
<b>LastMarkerColor</b>	このプロパティを設定すると、スパークラインの最後のデータポイントのマーカ

	の色を指定できます。
<b>HighMarkerColor</b>	このプロパティを設定すると、スパークラインの最大のデータポイントのマーカーの色を指定できます。
<b>LowMarkerColor</b>	このプロパティを設定すると、スパークラインの最小のデータポイントのマーカーの色を指定できます。
<b>NegativeColor</b>	このプロパティを設定すると、スパークラインの負のデータポイントのマーカーの色を指定できます。

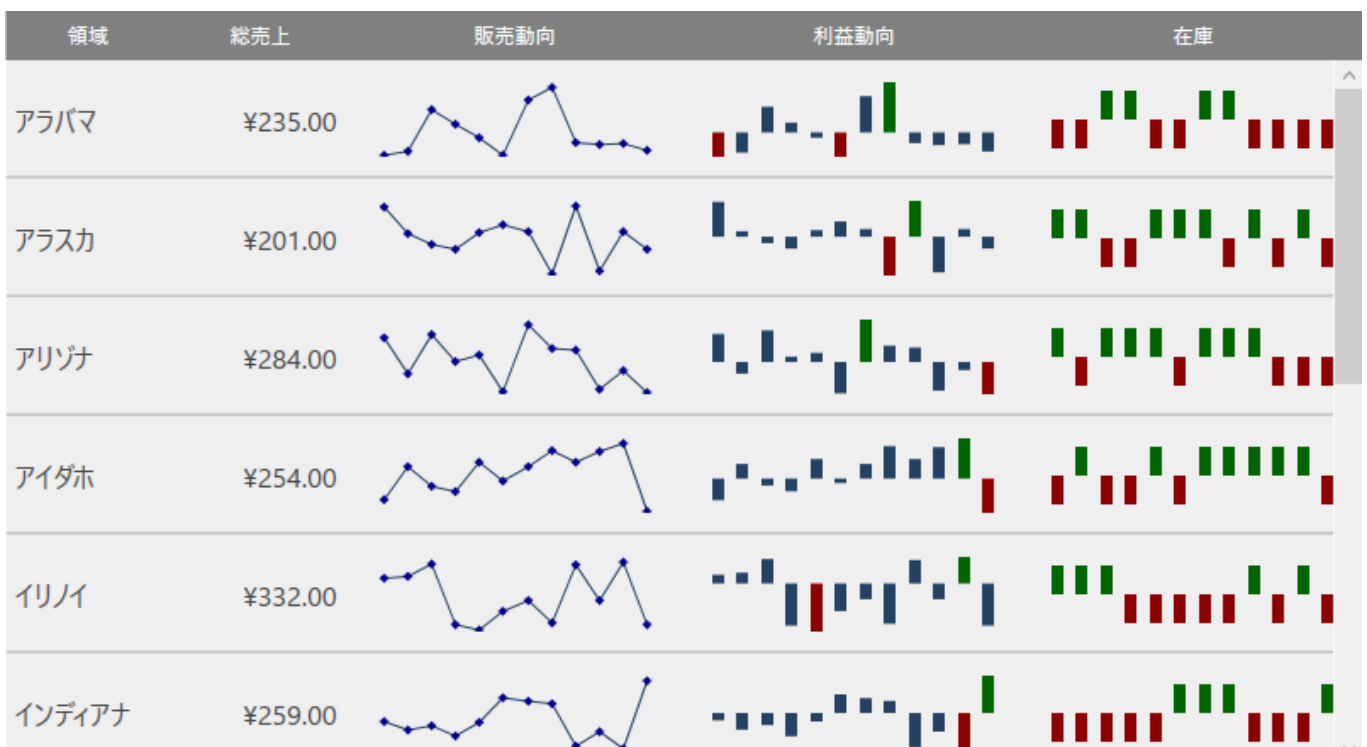
## タスク別ヘルプ

### データ管理コントロールでのスパークラインの使用

このセクションでは、ItemsControl、ListBox、データグリッドなどのデータ管理コントロールにスパークラインを追加する方法について説明します。最初に新しい WPF アプリケーションを作成し、5 列のフィールドを持つグリッド形式のレイアウトを表示するコードを XAML ビューで追加し、データを表示する相互作用ロジックをコードビューで追加します。

下の画像のように、グリッドに一連のスパークラインを追加するには、次の手順を実行します。

- **手順 1: アプリケーションの設定**
- **手順 2: XAML グリッドにデータとスパークラインを表示するコードの追加**
- **手順 3: アプリケーションの実行**



#### 手順 1: アプリケーションの設定

1. Visual Studio で WPF アプリケーションを作成します。
2. C1Sparkline コントロールを MainWindow にドラッグアンドドロップします。
3. XAML ビューを編集し、ComponentOne 名前空間をインクルードします。

```
XAML
```

```
xmlns:cl="http://schemas.componentone.com/winfx/2006/xaml"
```

4. グリッドタグを以下のコードで置き換えて、5 列のデータフィールド(領域、総売上、販売動向、利益動向、在庫)を表示する XAML グリッドを作成します。

XAML	copyCode
<pre>&lt;Grid&gt;   &lt;Grid.Resources&gt;     &lt;Style TargetType="TextBlock"&gt;       &lt;Setter Property="FontSize" Value="12" /&gt;     &lt;/Style&gt;   &lt;/Grid.Resources&gt;   &lt;Grid Width="800" Margin="10"&gt;     &lt;Grid.RowDefinitions&gt;       &lt;RowDefinition Height="30"/&gt;       &lt;RowDefinition /&gt;     &lt;/Grid.RowDefinitions&gt;     &lt;Grid Background="Gray"&gt;       &lt;Grid.ColumnDefinitions&gt;         &lt;ColumnDefinition Width="100"/&gt;         &lt;ColumnDefinition Width="100"/&gt;         &lt;ColumnDefinition Width="200"/&gt;         &lt;ColumnDefinition Width="200"/&gt;         &lt;ColumnDefinition Width="200"/&gt;       &lt;/Grid.ColumnDefinitions&gt;       &lt;TextBlock Text="領域" Foreground="White"         HorizontalAlignment="Center" VerticalAlignment="Center"/&gt;       &lt;TextBlock Text="総売上" Grid.Column="1" Foreground="White"         HorizontalAlignment="Center" VerticalAlignment="Center"/&gt;       &lt;TextBlock Text="販売動向" Grid.Column="2" Foreground="White"         HorizontalAlignment="Center" VerticalAlignment="Center"/&gt;       &lt;TextBlock Text="利益動向" Grid.Column="3" Foreground="White"         HorizontalAlignment="Center" VerticalAlignment="Center"/&gt;       &lt;TextBlock Text="在庫" Grid.Column="4" Foreground="White"         HorizontalAlignment="Center" VerticalAlignment="Center"/&gt;     &lt;/Grid&gt;   &lt;/Grid&gt;   &lt;ScrollViewer Grid.Row="1"&gt;     &lt;ItemsControl x:Name="RegionSalesListBox" ItemsSource="{Binding Sales}" &gt;       &lt;ItemsControl.ItemTemplate&gt;         &lt;DataTemplate&gt;           &lt;Grid Background="#EFEFEF"&gt;             &lt;Grid.ColumnDefinitions&gt;               &lt;ColumnDefinition Width="100"/&gt;               &lt;ColumnDefinition Width="100"/&gt;               &lt;ColumnDefinition Width="200"/&gt;               &lt;ColumnDefinition Width="200"/&gt;               &lt;ColumnDefinition Width="200"/&gt;             &lt;/Grid.ColumnDefinitions&gt;             &lt;TextBlock Text="{Binding State}"               Foreground="#444444"               FontSize="16" VerticalAlignment="Center"               HorizontalAlignment="Left" Margin="5" /&gt;           &lt;/Grid&gt;         &lt;/DataTemplate&gt;       &lt;/ItemsControl.ItemTemplate&gt;     &lt;/ItemsControl&gt;   &lt;/ScrollViewer&gt; &lt;/Grid&gt;</pre>	

```

        <TextBlock Text="{Binding TotalSalesFormatted}"
        Grid.Column="1"
        FontSize="16" Foreground="#444444"
        VerticalAlignment="Center"
        HorizontalAlignment="Right" Margin="5"/>
        <c1:C1Sparkline Data="{Binding Data}"
Grid.Column="2"
        Height="50" Margin="10" ShowMarkers="True"
        MarkersColor="DarkBlue"/>
        <c1:C1Sparkline Data="{Binding Data}"
        SparklineType="Column"
        Grid.Column="3" Height="50" Margin="10"
        ShowHigh="True"
        ShowLow="True" LowMarkerColor="DarkRed"
        HighMarkerColor="DarkGreen"/>
        <c1:C1Sparkline Data="{Binding Data}"
        SparklineType="Winloss"
        Grid.Column="4" Height="40" Margin="10"
        SeriesColor="DarkGreen" NegativeColor="DarkRed"
        ShowNegative="True"/>
        <Border Grid.ColumnSpan="6"
VerticalAlignment="Bottom"
        HorizontalAlignment="Stretch"
        BorderThickness="1" BorderBrush="#CCCCCC" />
    </Grid>
</DataTemplate>
</ItemsControl.ItemTemplate>
</ItemsControl>
</ScrollViewer>
</Grid>
</Grid>

```

## 手順 2: XAML グリッドにデータとスパークラインを表示するコードの追加

1. MainWindow.xaml.cs ファイルのコードビューに切り替えます。
2. 次の import 文を追加します。
3. XAML の相互作用ロジックを次のように編集して、グリッドにデータとスパークラインを表示します。

## 手順 3: アプリケーションの実行

1. アプリケーションをデバッグして、エラーがないかどうかを確認します。
2. [F5] キーを押してアプリケーションを実行し、出力を確認します。