Gauges for UWP

2018.03.07 更新

グレープシティ株式会社

Gauges for UWP

目次

Gauges for UWP

モダンな外観を持つインタラクティブなゲージにより、ダッシュボードに輝きを与えます。**Gauges for UWP** には、データの視覚化とビジネスダッシュボードの機能を強化するための複数のゲージコントロールが含まれます。これらのコントロールは、Windows タブレットでデータを魅力的に表示する方法を提供します。

主な特長

Gauges for UWP には、次の主な特長があります。

- **7つのゲージコントロール**
  Gauges for UWP には、形状の異なる7つのコントロールが含まれます。データに最も適したゲージを選択してください。
  - **C1RadialGauge**
  - **C1LinearGauge**
  - **C1Knob**
  - **C1RegionKnob**
  - **C1RulerGauge**
  - **C1SpeedometerGauge**
  - **C1VolumeGauge**
  - Interactive Gauges
- **インタラクティブゲージ**
  C1Knob コントロールでは、エンドユーザーがポインタを特定の値にドラッグできます。カスタマイズ可能な領域が表示されるユニークな C1RegionKnob もあります。インタラクティブなゲージは、テキストベースのエディタやスライダの代わりに使用でき、魅力的なユーザーエクスペリエンスを提供します。
- **目盛りマークおよびラベル**
  XAML またはコードでマークとラベルを定義します。単純なプロパティを使用して、マークとラベルの間隔、位置、および外観をカスタマイズします。ゲージのラベルに書式設定を適用します。たとえば、標準の書式文字列を使用して、通貨やパーセンテージなどの書式をラベルに指定します。
- **範囲**
  特定の範囲の値が注目されるように、ゲージに色付き範囲を追加します。単純なプロパティを使用して、開始点、終了点、位置、サイズ、および外観をカスタマイズします。開始幅と終了幅を指定して非直線型の範囲を作成することで、値の変化を表したり、見やすいゲージを作成することができます。
- **ポインタのカスタマイズ**
  単純なプロパティを使用して、ポインタおよびポインタキャップの外観と位置をカスタマイズします。
- **スケールのカスタマイズ**
  単純なプロパティを使用して、ゲージスケールの開始角度と移動角度を設定します。**Gauges for UWP** は、対数スケールもサポートします。
- **オフモードのサポート**
  値がない場合は、範囲外にオフポジションを設定できます。
- **ClearStyle を使用して簡単に色を変更する**
  Gauges for UWP は、コントロールのテンプレートを変更することなくコントロールの色を簡単に変更できる ComponentOne ClearStyle 技術をサポートします。Visual Studio で色のプロパティをいくつか設定するだけで、ゲージの外観を簡単に変更できます。

クイックスタート

このクイックスタートガイドは、**Gauges for UWP** を初めて使用するユーザーのために用意されています。このクイックスタートでは、Visual Studio で新しいプロジェクトを作成し、アプリケーションに **Gauges for UWP** コントロールを追加して、コントロールの外観と動作をカスタマイズします。

C1RadialGauge、**C1LinearGauge**、および **C1Knob** コントロールを含むアプリケーションを作成します。実行時にユーザーがスライダの値を変更すると、ゲージコントロールの値も変更されます。

手順1：アプリケーションの設定

この手順では、Visual Studioでアプリケーションを作成し、**StackPanel** パネルを追加してコントロールのレイアウトをカスタマイズします。

プロジェクトを設定するには、次の手順に従います。

1. Visual Studio で、[**ファイル**]→[**新規作成**]→[**プロジェクト**]を選択します。
2. [**新しいプロジェクト**]ダイアログボックスで、左ペインの言語を展開し、言語の下で[**Windows ストア**]を選択し、テンプレートリストで[**新しいアプリケーション (XAML)**]を選択します。プロジェクトの名前を入力し、[**OK**]をクリックしてプロジェクトを作成します。
   MainPage.xaml ページが表示され、<Grid> タグと </Grid> タグの間にカーソルが置かれます。
3. ツールボックスに移動し、[**StackPanel**]アイコンをダブルクリックして、**MainPage.xaml** にパネルを追加します。
4. x:Name="sp1" Width="Auto" Height="Auto" Orientation="Vertical" HorizontalAlignment="Center" VerticalAlignment="Center" を <StackPanel> タグに追加します。次のようになります。

   マークアップ

   ```
   <StackPanel x:Name="sp1" Width="Auto" Height="Auto" Orientation="Vertical"
   HorizontalAlignment="Center" VerticalAlignment="Center"></StackPanel>
   ```

   これで、パネル内の要素は、中央で縦方向に配置されて表示されます。
5. プロジェクトの XAML ウィンドウで、カーソルを <StackPanel> タグと </StackPanel> タグの間に置きます。
6. ツールボックスに移動し、[**StackPanel**]アイコンをダブルクリックして、既存の **StackPanel** にパネルを追加します。
7. x:Name="sp2" Width="Auto" Height="Auto" Orientation="Horizontal" HorizontalAlignment="Center" VerticalAlignment="Center"を <StackPanel> タグに追加します。次のようになります。

   マークアップ

   ```
   <StackPanel x:Name="sp2" Width="Auto" Height="Auto" Orientation="Horizontal"
   HorizontalAlignment="Center" VerticalAlignment="Center"></StackPanel>
   ```

   これで、パネル内の要素は、中央で横方向に配置されて表示されます。

これで、新しい UWP スタイルのプロジェクトが作成され、アプリケーションが正しく設定されました。次の手順では、いくつかの **Gauges for UWP** コントロールをアプリケーションに追加し、これらのコントロールをカスタマイズします。

手順2：コントロールの追加

この手順では、プロジェクトに **C1RadialGauge**、**C1LinearGauge**、および **C1Knob** コントロールを追加して、アプリケーションを設定します。

これらのゲージコントロールをアプリケーションに追加するには、次の手順に従います。

1. プロジェクトの XAML ウィンドウで、カーソルを <StackPanel x:Name="sp2"> タグと </StackPanel> タグの間に置きます。
2. ツールボックスに移動し、[**C1Knob**]アイコンをダブルクリックして、**StackPanel** にコントロールを追加します。これで、参照と XAML 名前空間が自動的に追加されます。
3. x:Name="c1kb1" を <Gauge:C1Knob> タグに追加して、コントロールに名前を付けます。次のようになります。

マークアップ

```
<Gauge:C1Knob x:Name="c1kb1">
```

それに一意の識別子を付けると、コードでそのコントロールにアクセスできるようになります。

4. `<Gauge:C1Knob>` タグに `Width="150"` を追加して、コントロールをサイズ変更します。次のようになります。

マークアップ

```
<Gauge:C1Knob x:Name="c1kb1" Width="150">
```

アプリケーションを実行すると、このコントロールは少し小さく表示されます。

5. `<Gauge:C1Knob>` タグに `Margin="5"` を追加して、コントロールにマージンを追加します。次のようになります。

マークアップ

```
<Gauge:C1Knob x:Name="c1kb1" Width="150" Margin="5">
```

これで、**C1Knob** とページに追加する他のコントロールの間にスペースが追加されます。

6. `Minimum="0" Maximum="100"` を `<Gauge:C1Knob>` タグに追加して、最小値と最大値を設定します。次のようになります。

マークアップ

```
<Gauge:C1Knob x:Name="c1kb1" Width="150" Margin="5" Minimum="0" Maximum="100">
```

これは、ノブで設定できる最高値と最低値を決定します。

7. プロジェクトの XAML ウィンドウで、カーソルを `</Gauge:C1Knob>` タグと `</StackPanel>` タグの間に置きます。
8. ツールボックスに移動し、[**C1RadialGauge**]アイコンをダブルクリックして、**StackPanel** にコントロールを追加します。
9. `x:Name="c1rg1" Margin="5" Minimum="0" Maximum="100" Height="300"` を `<Gauge:C1RadialGauge>` タグに追加して、コントロールをカスタマイズします。次のようになります。

マークアップ

```
<Gauge:C1RadialGauge x:Name="c1rg1" Margin="5" Minimum="0" Maximum="100" Height="300"
Value="100" StartAngle="0" SweepAngle="300"></Gauge:C1RadialGauge>
```

これで、**C1RadialGauge** の名前が指定され、コントロールがサイズ変更され、コントロールの最小値と最大値が設定されます。

10. `<Gauge:C1RadialGauge>` タグと `</Gauge:C1RadialGauge>` タグの間に次のマークアップを追加して、ゲージの外観を変更します。

マークアップ

```
<Gauge:C1GaugeRange To="40" Location="0.8" Fill="#088080" Width="0.2" Opacity="0.6" />
<Gauge:C1GaugeRange From="75" Fill="#088080" Location="0.9" EndWidth="0.2" Opacity="0.3" />
<Gauge:C1GaugeMark Interval="20" />
<Gauge:C1GaugeMark Interval="10" />
<Gauge:C1GaugeMark Interval="1" />
<Gauge:C1GaugeLabel Interval="20" Alignment="In" AlignmentOffset="10" FontSize="16" />
```

これで、ゲージの範囲と目盛りマークの外観が設定されます。

11. プロジェクトの XAML ウィンドウで、カーソルを最初と2番目の `</StackPanel>` タグの間に置きます。
12. ツールボックスに移動し、[**Slider**]アイコンをダブルクリックして、**StackPanel** に標準コントロールを追加します。
13. `x:Name="s1" Height="400" Minimum="0" Maximum="100" ValueChanged="s1_ValueChanged_1" Orientation="Vertical"` を `<Slider>` タグに追加して、コント

ロールをカスタマイズします。次のようになります。

マークアップ
<Slider x:Name="s1" Height="400" Minimum="0" Maximum="100" ValueChanged="s1_ValueChanged_1" Orientation="Vertical"/>

これで、**Slider** の名前が指定され、コントロールがサイズ変更され、最小値と最大値が設定されます。イベントハンドラのコードは、この後の手順で追加します。

14. プロジェクトの XAML ウィンドウで、カーソルを `<Slider />` タグと `</StackPanel>` タグの間に置きます。
15. ツールボックスに移動し、[**C1LinearGauge**]アイコンをダブルクリックして、**StackPanel** にコントロールを追加します。
16. `x:Name="c1lg1" Minimum="0" Maximum="100" Width="120" Height="500"` を `<Gauge:C1LinearGauge>` タグに追加して、コントロールをカスタマイズします。次のようになります。

マークアップ
<Gauge:C1LinearGauge x:Name="c1lg1" Minimum="0" Maximum="100" Width="120" Height="500" Orientation="Vertical" XAxisLocation="0.05" XAxisLength="0.9" YAxisLocation="0.2"> </Gauge:C1LinearGauge>

これで、**C1LinearGauge** コントロールの名前が指定され、コントロールがサイズ変更され、最小値と最大値が設定されます。

17. `<Gauge:C1LinearGauge>` タグと `</Gauge:C1LinearGauge>` タグの間に次のマークアップを追加して、ゲージの外観を変更します。

マークアップ
<Gauge:C1GaugeMark Interval="20" /> <Gauge:C1GaugeMark Interval="10" /> <Gauge:C1GaugeMark Interval="2" /> <Gauge:C1GaugeLabel Interval="20" Format="n0" Alignment="Out" AlignmentOffset="30" FontSize="16"/> <Gauge:C1GaugeRange To="40" Location="0" Fill="#088080" Width="0.2" Opacity="0.2"/> <Gauge:C1GaugeRange From="40" To="80" Location="0" Fill="#088080" Width="0.2" Opacity="0.4"/> <Gauge:C1GaugeRange From="80" To="100" Location="0" Fill="#088080" Width="0.2" Opacity="0.6"/>

これで、ゲージの範囲と目盛りマークの外観が設定されます。

いくつかの **Gauges for UWP** コントロールをアプリケーションに追加し、これらのコントロールをカスタマイズしました。これで、アプリケーションのユーザーインタフェースを正しく設定できました。次の手順では、コードをアプリケーションに追加します。

手順3：アプリケーションへのコードの追加

前の手順では、新しい UWP スタイルのプロジェクトを作成し、アプリケーションにいくつかの **Gauges for UWP** コントロールを追加しました。この手順では、アプリケーションにコードを追加してカスタマイズします。

次の手順に従います。

1. [**表示**]→[**コード**]を選択してコードビューに切り替えます。
2. 次の imports 文をページの先頭に追加します。

▶ Visual Basic コードの書き方

Visual Basic
Imports C1.Xaml Imports C1.Xaml.Gauge

▶ C# コードの書き方

```
C#
using C1.Xaml;
using C1.Xaml.Gauge;
```

3. **s1_ValueChanged_1** イベントハンドラに、ゲージとスライダコントロールの値を設定するコードを追加します。次のようになります。

▶ Visual Basic コードの書き方

```
Visual Basic
Private Sub s1_ValueChanged_1(sender As Object, e As  RangeBaseValueChangedEventArgs)
    Me.c1lg1.Value = Me.s1.Value
    Me.c1rg1.Value = Me.s1.Value
    Me.c1kb1.Value = Me.s1.Value
End Sub
```

▶ C# コードの書き方

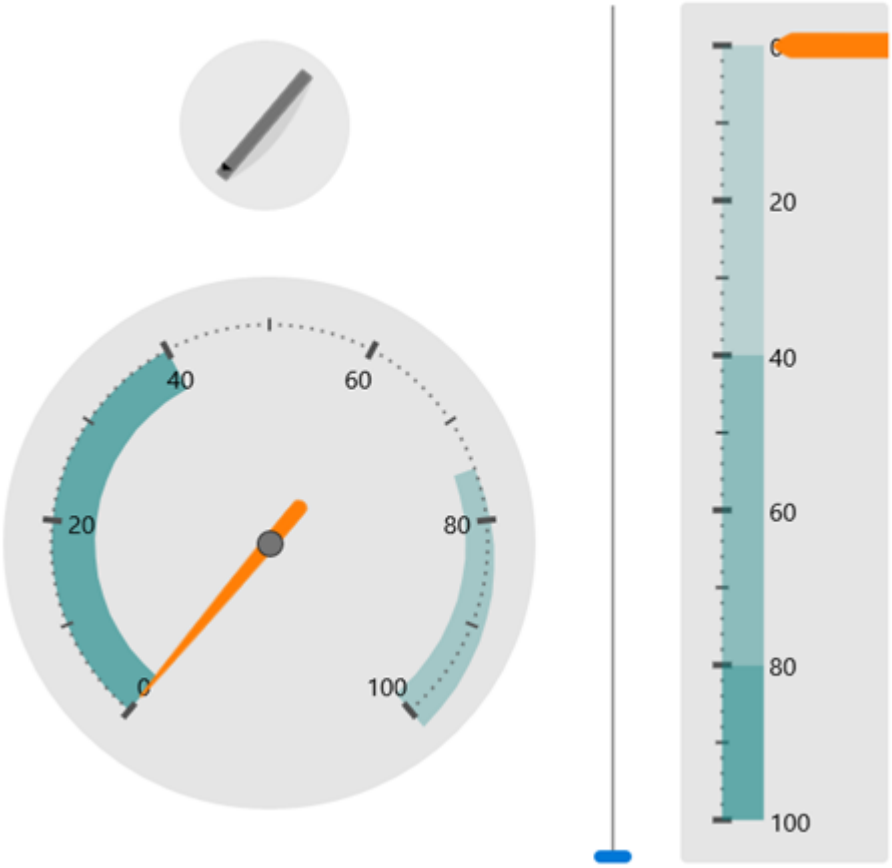
```
C#
private void s1_ValueChanged_1(object sender, RangeBaseValueChangedEventArgs e)
{
    this.c1lg1.Value = this.s1.Value;
    this.c1rg1.Value = this.s1.Value;
    this.c1kb1.Value = this.s1.Value;
}
```

この手順では、アプリケーションにコードを追加しました。次の手順では、アプリケーションを実行し、実行時の操作を確認します。

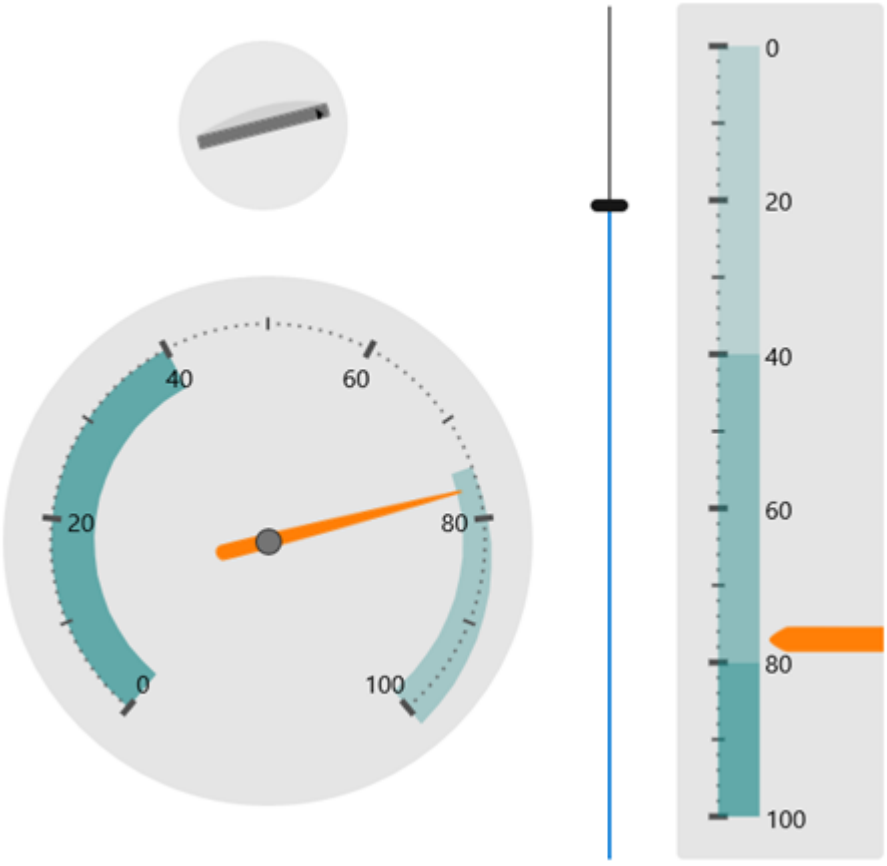
手順4：アプリケーションの実行

これまでに、UWP スタイルアプリケーションを作成し、外観と動作をカスタマイズしたので、次にアプリケーションを実行します。アプリケーションを実行し、**Gauges for UWP** の実行時の動作を確認するには、次の手順に従います。

1. [**デバッグ**]メニューから[**デバッグ開始**]を選択し、実行時にアプリケーションがどのように表示されるかを確認します。アプリケーションは次の図のように表示されます。



2. スライダのサムボタンをクリックしてドラッグします。**C1Knob**、**C1RadialGauge**、および **C1LinearGauge** コントロールの値が変化することを確認します。



おめでとうございます。これで **Gauges for UWP** クイックスタートは完了です。C1RadialGauge、C1LinearGauge、および C1Knob コントロールを使用するアプリケーションを作成し、アプリケーションの実行時機能をいくつか確認することができました。

ゲージコントロールが便利な理由

ゲージは1つの値を示すだけです。ゲージではなく単純なラベルを使用して値を表示することもできるのに、なぜゲージコントロールを使用する必要があるのでしょうか。

ゲージには範囲を表示することもできるため、ユーザーは現在の値が大きいか、小さいか、それとも中間であるかを即座に判断することができます。このため、ゲージの方がわかりやすくて便利です。2つのラベルを追加して範囲と現在値を表示することもできますが、こうすると、わかりにくいユーザーインタフェースになります。このため、多くのアプリケーションでは、進行状況を表示するために、ラベルではなく、単純な直線型ゲージの進捗状況インジケータを使用しています。

ゲージは、単純なラベル（またはスライダやスクロールバー）より視覚的にわかりやすく、アプリケーションの価値を高めます。

しかし、デザイナに XAML で魅力的なゲージを作成してもらい、要素をアニメーション表示して現在値を示すこともできるのに、ゲージコントロールを使用する理由があるのでしょうか。なぜ、コントロールを使用するのでしょうか。

これには、いくつかの理由があります。まず、誰もが優れたデザイナというわけではなく、また、優れたデザイナに依頼できるとも限りません。次に、アプリケーションに必要なゲージが1つだけではないこともあります。さまざまな範囲に渡る値を表示するために、複数のゲージが必要になる可能性があります。アプリケーションを記述するときには、実際の範囲（現四半期の最大売上値など）すらわかっていない場合もあります。

ゲージコントロールを使用すれば、手作業で XAML のコードを記述しなくても、データに基づいて、プログラムによって柔軟に範囲を調整できます。

C1RadialGauge の使い方

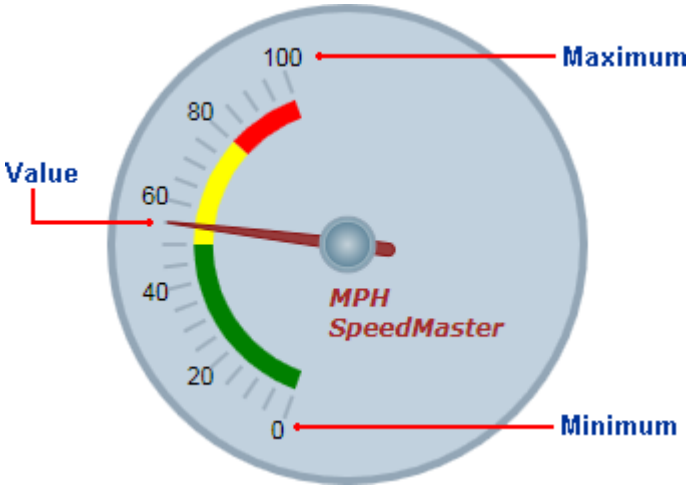
C1RadialGauge は、回転型のポインタを使用して、曲線目盛りに沿って値を表示します。**C1RadialGauge** コントロールは、回転型のポインタを使用して値を表示します。値は **Value** プロパティで表されます。範囲は **Minimum** プロパティと **Maximum** プロパティで定義されます。C1RadialGauge コントロールは、典型的なスピードメーターに似ています。

C1RadialGauge を作成して使用するには、通常、次の手順に従います。

1. C1RadialGauge コントロールを作成し、メインプロパティの Minimum、Maximum、**StartAngle**、および **SweepAngle** を設定します。
2. **C1GaugeMark** および **C1GaugeLabel** デコレータを追加してスケールを表示します。各要素には、ラベル、目盛りマーク、またはその両方を表示できます。
3. オプションで、**C1GaugeRange** デコレータを追加して、スケールの一部を強調表示します。これらの範囲は、通常、小さすぎる値、適正値、または大きすぎる値を示すために使用されます。**Value** プロパティが変化すると範囲が自動的に移動するように、範囲を動的にすることもできます。
4. オプションで、XAML テンプレートを使用してゲージをカスタマイズします。
5. Value プロパティを設定して、初めに表示される値を指定します。

C1RadialGauge の値

C1RadialGauge コントロールの **Minimum**、**Maximum**、および **Value** プロパティを使用すると、有効な範囲およびその範囲内で選択された値を指定できます。



Minimum および **Maximum** プロパティは、ゲージに表示される値の範囲を指定します。たとえば、温度計には -40 ～ 100 度の範囲、スピードメーターには時速0～ 140 マイルの範囲を割り当てることができます。この範囲は、Minimum および Maximum プロパティ(**double** 型)を使用して指定されます。C1RadialGauge コントロールのデフォルトの範囲は0～ 100 です。

Value プロパティは、ゲージの現在の値を指定します。**C1RadialGauge** コントロールでは、C1GaugePointer 要素がこの値を指し示すことによってこれが視覚的に示されます。C1RadialGauge コントロールのデフォルトの Value は 50 です。

C1RadialGauge の StartAngle および SweepAngle

範囲を定義したら、**Minimum** および **Maximum** 値に対応する角度を指定する必要があります。**StartAngle** は、**Value** プロパティが範囲の Minimum 値に設定されているときのポインタの位置を定義します。**SweepAngle** は、Value プロパティが範囲の Maximum 値に設定されているときの StartAngle からの回転角度を指定します。

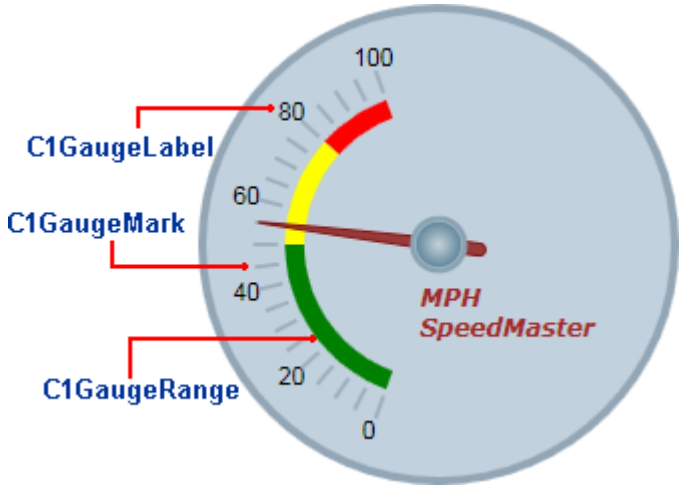
すべての角度は、コントロールの上中央から時計回りに度単位で指定されます。角度には負の値も指定できますが、SweepAngle の絶対値が 360 度を超えることはできません。StartAngle のデフォルト値は -140、SweepAngle のデフォルト値は 280 です。

次の図は、StartAngle プロパティと SweepAngle プロパティのさまざまな値による違いを示します。

 StartAngle = 0 SweepAngle = 90	 StartAngle = 0 SweepAngle = -90	 StartAngle = 45 SweepAngle = 270
 StartAngle = -160 SweepAngle = 180	 StartAngle = -160 SweepAngle = -180	 StartAngle = -140 SweepAngle = 280

C1RadialGauge のデコレータ

C1RadialGauge コントロールには、デフォルトでは青灰色の背景とポインタのみが表示されます。ただし、多くのアプリケーションでは、現在の値がいくつで、その値がゲージの範囲内のどこに位置するかがわかるように、ラベルと目盛りマークがスケールとして表示されます。それには、**C1GaugeMark**、**C1GaugeLabel**、および **C1GaugeRange** 要素をゲージの **Decorators** コレクションに追加します。



これらのデコレータはスケール上の特定の位置に表示され、その位置は **From**、**To**、および **Interval** プロパティの値で決定されます。

上の画像では、1つの C1GaugeMark 要素と1つの C1GaugeLabel 要素が表示されています。

```
<!-- ラベルマークの追加 -->
<Gauge:C1GaugeLabel From="0" To="100" Interval="20" Location="1.1"/>

<!-- 目盛りマークの追加 -->
<Gauge:C1GaugeMark From="0" To="100" Interval="5" Location=".9"/>
```

C1GaugeLabel 要素は、スケールに沿って 0 から 100 までの値のラベルを表示します。この C1GaugeMark 要素は、5きざみに目盛りマークを表示します。

スケールを表示するほかに、スケール範囲の一部を強調表示することができます。たとえば、赤色のマーカーを追加し、その範囲は値が小さすぎる(売上)ことや、大きすぎる(費用)ことを示すことができます。それには、いくつかの C1GaugeRange 要素をゲージの Decorators コレクションに追加します。

上の画像では、3つの C1GaugeRange 要素が表示されています。

```
<!-- 3つの色付き範囲の追加 -->
<Gauge:C1GaugeRange From="80" To="100" Location="0.7" Fill="Red" />
<Gauge:C1GaugeRange From="50" To="80" Location="0.7" Fill="Yellow" />
<Gauge:C1GaugeRange From="0" To="50" Location="0.7" Fill="Green" />
```

これらの C1GaugeRange 要素は、それぞれ赤色、黄色、および緑色の範囲を示します。各 C1GaugeRange 要素は、スケールに沿った1つの曲線型の帯として表示されます。帯の色は **Fill** プロパティによって決定され、位置は From および To プロパティによって決定されます。帯の太さは、**StrokeThickness** プロパティを使用して制御できます。

C1RadialGauge デコレータの場所

各デコレータ要素には、要素が表示される位置を指定する **Location** プロパティがあります。このプロパティには、0(ゲージの中心)から1(ゲージの外縁)までを指定できます。ゲージコントロールには、すべてのデコレータの配置を制御する **Radius** プ

ロパティもあり、これも0から1までを指定できます。半径プロパティのデフォルト値は 0.8 です。この場合は、すべてのデコレータがコントロール内に表示されます。

C1RadialGauge デコレータのサンプルでは、**C1GaugeLabel** の Location プロパティが 1.1 に設定されています。これにより、ラベルはゲージの外側にオフセットされて表示されます。Radius プロパティは 0.8 に設定されているため、ラベルはコントロール内に描画されます（この場合、ラベルの実際の位置は、1.1 * 0.8 = 0.88 と計算できます）。

次の図は、サンプルゲージの **C1GaugeMark** および C1GaugeLabel 要素にさまざまな Location プロパティを適用した場合の違いを示します。

 **C1GaugeLabel.Location = 1.1** **C1GaugeMark.Location = 1**	 **C1GaugeLabel.Location = 1** **C1GaugeMark.Location = 1**	 **C1GaugeLabel.Location = 0.6** **C1GaugeMark.Location = 1**
 **C1GaugeLabel.Location = 1.12** **C1GaugeMark.Location = 1.05**	 **C1GaugeLabel.Location = 1** **C1GaugeMark.Location = 1.3**	 **C1GaugeLabel.Location = 1.1** **C1GaugeMark.Location = 1.4**

Location に1より大きな値を指定すると、ラベルやマークがゲージ本体の外側に描画されます。

C1GaugeMark 要素は、必要な数だけ指定できます。たとえば、0～ 60 分の範囲を持つ時計型のゲージを作成できます。その場合は、1つの C1GaugeLabel 要素と2つの C1GaugeMark 要素を使用します。

- 4つの「主な」時間（12、3、6、9）のラベルを表示する 15 分間隔の C1GaugeLabel。
- 1時間を表す5分間隔の C1GaugeMark。
- 1分を表す1分間隔の C1GaugeMark。

ラベルと目盛りマークは、**Template** プロパティを使用してカスタマイズすることもできます。

C1RadialGauge デコレータの値連結

範囲は、静的な値に限りません。**ValueBinding** プロパティを使用して、範囲の開始位置または終了位置をゲージに表示された現在の値に連結できます。たとえば、次のコードは、速度が時速 80 マイルを超えた場合にのみ、赤色の範囲を表示します。

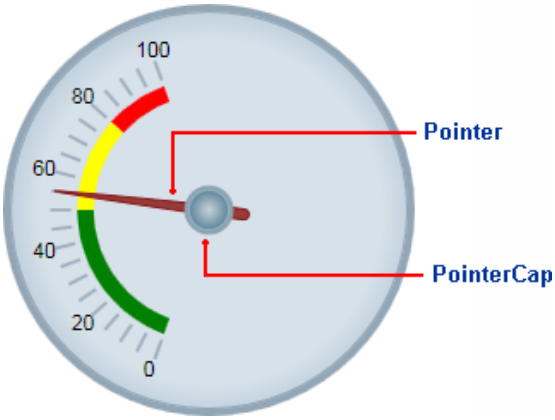
```
<!-- 3つの色付き範囲の追加 -->
<Gauge:C1GaugeRange From="80" ValueBinding="To" Location="0.7" Background="Red" />
<Gauge:C1GaugeRange From="50" To="80" Location="0.7" Fill="Yellow" />
<Gauge:C1GaugeRange From="0" To="50" Location="0.7" Fill="Green" />
```

次の図は、**Value** プロパティの変化に伴う表示の変化を示します。

		
Value = 55	**Value = 90**	**Value = 100**

C1RadialGauge の Pointer および PointerCap

C1RadialGauge コントロールには、選択されている **Value** をコントロールに示すポインタがあります。ポインタは、実際には C1GaugePointer 要素と **PointerCap** 要素で構成されます。



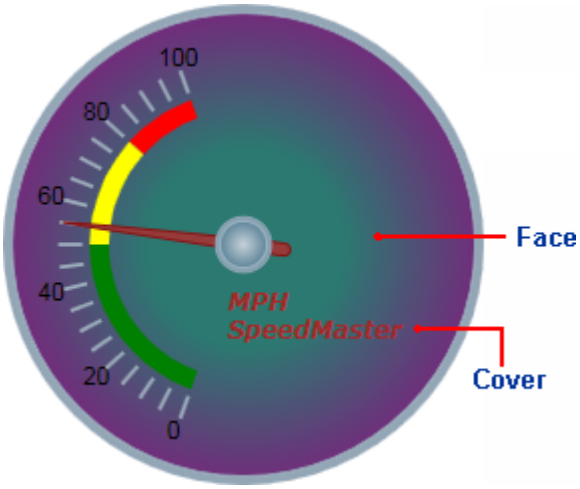
PointerOrigin プロパティは、ポインタの場所を設定します。ポインタの場所は、このプロパティを設定することでカスタマイズできます。たとえば、ポイント(0, 0)は、コントロールの左上隅を示し、ポイント(0.5, 0.5)は、コントロールの中心を示します。**Location** プロパティは、C1GaugePointer 要素の相対的な場所を設定します。

C1GaugePointer 要素は、デフォルトでは茶色の先細の要素として表示されますが、PointerFill、**PointerLength**、**PointerOffset**、**PointerStroke**、**PointerStrokeThickness**、**PointerStyle**、**PointerVisibility**、**PointerWidth** などのプロパティを設定することで、C1GaugePointer 要素の外観をカスタマイズできます。

PointerCap 要素は、デフォルトでは灰色の円として表示されますが、**PointerCap**、**PointerCapFill**、**PointerCapStroke**、**PointerCapStrokeThickness**、**PointerCapStyle** などのプロパティを設定することで、PointerCap 要素の外観をカスタマイズできます。**PointerCapSize** テンプレートを編集することもできます。

C1RadialGauge の Face および Cover

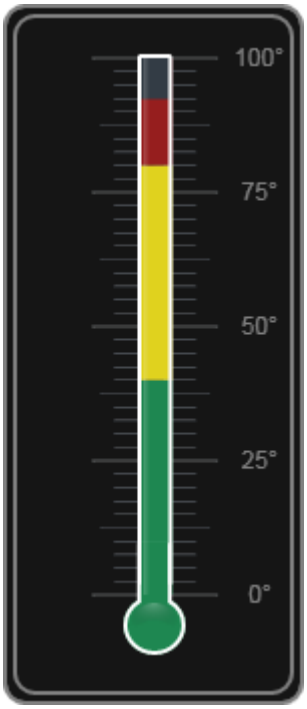
時計と同様に、C1RadialGauge コントロールには、**Face** と **Cover** が含まれています。Face は背景の上に表示され、ポインタなどのデコレータの下に表示されます。Cover は、時計を覆うガラスカバーのように、他のすべての要素の上に表示されます。たとえば、次の図では、Face がグラデーション付きでゲージ内の要素の下に表示されています。Cover にはテキストが表示され、Face と他のすべての要素の上に表示されています。



Cover の外観をカスタマイズするには、**CoverTemplate** を使用します。Face の外観をカスタマイズするには、**FaceTemplate** を使用します。

C1LinearGauge の使い方

C1LinearGauge コントロールのオブジェクトモデルは、**C1RadialGauge** のオブジェクトモデルとほとんど同じです。**C1LinearGauge** は、直線型のポインタを使用して、値を直線目盛りに沿って表示します。これは、典型的な温度計に似ています。

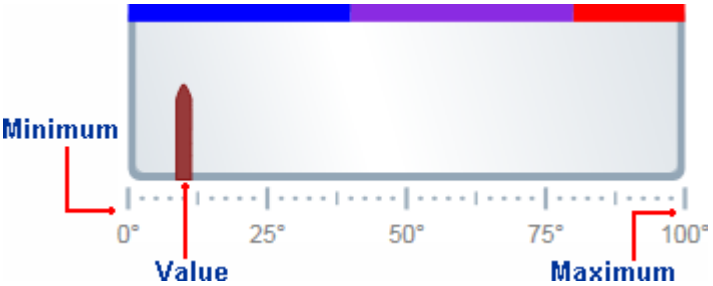


C1LinearGauge コントロールを作成して使用する手順は、C1RadialGauge で説明した手順と同じです。

1. C1LinearGaugeコントロールを作成し、メインプロパティの **Minimum**、**Maximum**、および **Orientation** を設定します。
2. C1GaugeMark デコレータを追加してスケールを表示します。各 C1GaugeMark 要素には、ラベル、目盛りマーク、またはその両方を表示できます。
3. オプションで、**C1GaugeRange** デコレータを追加して、スケールの一部を強調表示します。これらの範囲は、通常、小さすぎる値、適正値、または大きすぎる値を示すために使用されます。**Value** プロパティが変化すると範囲が自動的に移動するように、範囲を動的にすることもできます。
4. オプションで、XAML テンプレートを使用してゲージをカスタマイズします。
5. **Value** プロパティを設定して、初めに表示される値を指定します。

C1LinearGauge の値

C1LinearGauge コントロールの **Minimum**、**Maximum**、および **Value** プロパティを使用すると、有効な範囲およびその範囲内で選択された値を指定できます。



Minimum および Maximum プロパティは、ゲージに表示される値の範囲を指定します。たとえば、温度計には -40 ～ 100 度の範囲、スピードメーターには時速0～ 140 マイルの範囲を割り当てることができます。この範囲は、Minimum および

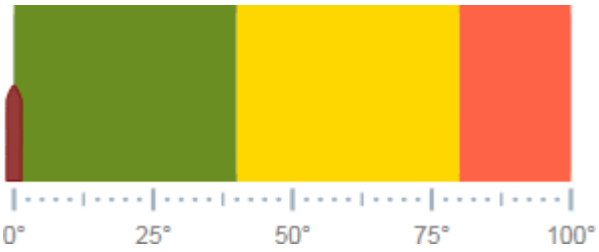
Maximum プロパティ(**double** 型)を使用して指定されます。C1LinearGauge コントロールのデフォルトの範囲は0～ 100 です。

Value プロパティは、ゲージの現在の値を指定します。C1LinearGauge コントロールでは、C1GaugePointer 要素がこの値を指し示すことによってこれが視覚的に示されます。C1LinearGauge コントロールのデフォルトの Value は0です。上の図では、Value は 10 に設定されています。

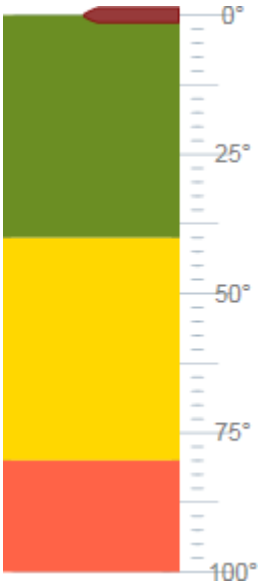
C1LinearGauge の方向

C1LinearGauge コントロールには、円形ゲージで使用される **StartAngle** および **SweepAngle** プロパティはありません。代わりに、ゲージを縦型または横型として作成するために使用する **Orientation** プロパティがあります。

デフォルトでは、Orientation プロパティは **Horizontal** に設定され、ゲージはアプリケーション内で横向きに表示されます。



Orientation プロパティを **Vertical** に設定することで、縦型のゲージを作成できます。



デフォルトでは、縦の C1LinearGauge コントロールは、上から下に向けてスケールを表示します(上の例では0～ 100)。スケールの方向を逆にするには、次のプロパティを設定する必要があります。

- **XAxisLocation** を **1** に設定します
- **XAxisLength** を **-1** に設定します

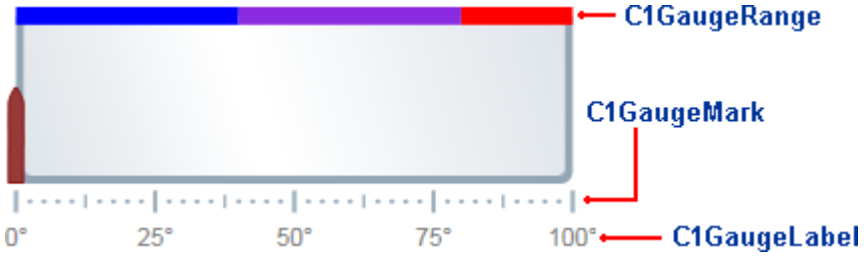
スケールが逆になるため、100 がゲージの上端、0が下端になります。

直線型ゲージには、さまざまな応用があります。たとえば、「**C1LinearGauge の使い方**」に表示されているゲージのように、縦の直線型ゲージを温度計として使用できます。

C1LinearGauge のデコレータ

デフォルトでは、**C1LinearGauge** コントロールには、単純な横の直線型ゲージのみが表示されます。ただし、多くのアプリケーションでは、現在の値がいくつで、その値がゲージの範囲内のどこに位置するかがわかるように、ラベルと目盛りマークがスケールとして表示されます。それには、**C1GaugeMark**、**C1GaugeLabel**、および **C1GaugeRange** 要素をゲージ

の **Decorators** コレクションに追加します。



これらのデコレータはスケール上の特定の位置に表示され、その位置は **From**、**To**、および **Interval** プロパティの値で決定されます。

上の画像では、3つの C1GaugeMark 要素と1つの C1GaugeLabel 要素が表示されています。

```
<!--目盛りマークの追加 -->
<Gauge:C1GaugeMark From="0" To="100" Interval="25" Location="1.1" />
<Gauge:C1GaugeMark From="0" To="100" Interval="12.5" Location="1.1" />
<Gauge:C1GaugeMark From="0" To="100" Interval="2.5" Location="1.1" />
<!--ラベルマークの追加 -->
<Gauge:C1GaugeLabel Location="1.3" Interval="25" Foreground="Gray" Alignment="Center" Format="0°" />
```

この C1GaugeLabel 要素は、スケールに沿って、0から 100 までの値に対して 25 きざみにラベルを表示します。C1GaugeMark 要素は、25、12.5、および 2.5 きざみに目盛りマークを表示します。

スケールを表示するほかに、スケール範囲の一部を強調表示することができます。たとえば、赤色のマーカーを追加し、その範囲は値が小さすぎる（売上）ことや、大きすぎる（費用）ことを示すことができます。それには、いくつかの C1GaugeRange 要素をゲージの Decorators コレクションに追加します。

上の画像では、3つの C1GaugeRange 要素が表示されています。

```
<!-- 3つの色付き範囲の追加 -->
<Gauge:C1GaugeRange Fill="Blue" To="40" Width=".1" />
<Gauge:C1GaugeRange Fill="BlueViolet" From="40" To="80" Width=".1" />
<Gauge:C1GaugeRange Fill="Red" From="80" To="100" Width=".1" />
```

これらの C1GaugeRange 要素は、青色、青紫色、および赤色の範囲を表示します。各 C1GaugeRange 要素は、スケールに沿った1つの曲線型の帯として表示されます。帯の色は **Fill** プロパティによって決定され、位置は From および To プロパティによって決定されます。帯の太さは、**Width** プロパティを使用して制御できます。

C1LinearGauge デコレータの場所

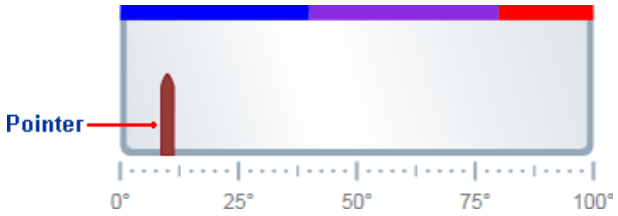
デコレータは、C1RadialGauge コントロールと同様に **C1LinearGauge** コントロールに対して配置されます。C1RadialGauge コントロールには、デコレータが表示される位置をゲージの中心からの距離で指定する Radius プロパティがありました。Radius プロパティには、0（ゲージの中心）から1（ゲージの外縁）までの値を指定できます。個々のデコレータは、**Location** プロパティで指定された量だけ Radius からオフセットされます。

C1LinearGauge には、Radius に似た **YAxisLocation** プロパティがあります。このプロパティには、0（ゲージの上端）から1（ゲージの下端）までの値を指定できます。個々のデコレータは、**Location** プロパティで指定された量だけ YAxisLocation からオフセットされます。

YAxisLocation プロパティのデフォルト値は0です。**C1GaugeMark** デコレータの Location のデフォルト値は1です（デフォルトでは、ゲージの下端に要素が表示されます）。C1GaugeRange デコレータの **Location** プロパティのデフォルト値は0です（デフォルトでは、ゲージの上端に要素が表示されます）。

C1LinearGauge の Pointer

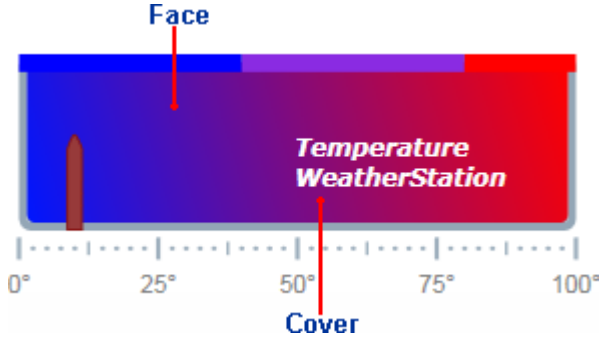
C1LinearGauge コントロールには、選択されている **Value** をコントロールに示すポインタがあります。ポインタは、C1GaugePointer 要素で構成されます。



C1GaugePointer 要素は、デフォルトでは茶色の先細の要素として表示されます
が、**PointerFill**、**PointerLength**、**PointerOffset**、**PointerStroke**、**PointerStrokeThickness**、**PointerStyle**、**PointerVisibility**、**PointerWidth** などのプロパティを設定することで、C1GaugePointer 要素の外観をカスタマイズできます。**Orientation** プロパティを **Vertical** または **Horizontal** に設定することで、C1GaugePointer 要素の表示方向をカスタマイズすることもできます。

C1LinearGauge の Face および Cover

時計や温度計と同様に、**C1LinearGauge** コントロールには、**Face** と **Cover** が含まれています。Face は背景の上に表示され、ポインタなどのデコレータの下に表示されます。Cover は、温度計を覆うガラスカバーのように、他のすべての要素の上に表示されます。たとえば、次の図では、Face がグラデーション付きでゲージ内の要素の下に表示されています。Cover にはテキストが表示され、Face と他のすべての要素の上に表示されています。



Cover の外観をカスタマイズするには、**CoverTemplate** を使用します。Face の外観をカスタマイズするには、**FaceTemplate** を使用します。

C1Knob の使い方

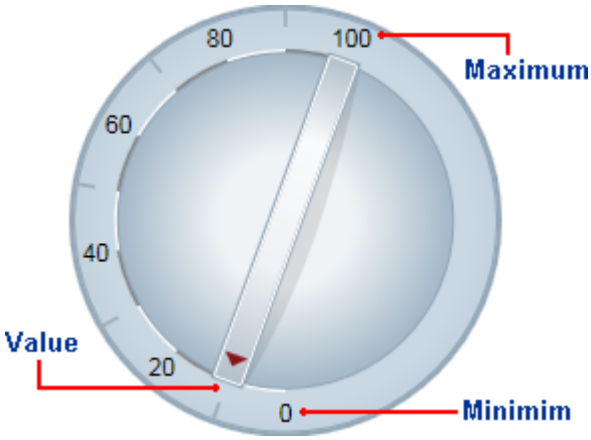
C1Knob コントロールは、C1RadialGauge コントロールを拡張して、ユーザーがポインタを回転して数値を選択できるようにします。たとえば、C1Knob は、音楽プレイヤーのボリュームつまみを真似る場合に最適です。デフォルトでは、C1Knob コントロールは次の図のようになります。

C1Knob コントロールを作成して使用する方法は、C1RadialGauge コントロールの作成方法に似ており、通常は同様の手順に従います。

1. C1Knob コントロールを作成し、メインプロパティの Minimum、Maximum、StartAngle、および SweepAngle を設定します。
2. **InteractionMode** プロパティを設定することで、ユーザーがノブを操作する方法を指定します。
3. C1GaugeMark および C1GaugeLabel デコレータを追加してスケールを表示します。各要素には、ラベル、目盛りマーク、またはその両方を表示できます。
4. オプションで、XAML テンプレートを使用してゲージをカスタマイズします。
5. Value プロパティを設定して、初めに表示される値を指定します。

C1Knob の値

C1Knob コントロールの Minimum、Maximum、および Value プロパティを使用すると、有効な範囲およびその範囲内で選択された値を指定できます。



Minimum および Maximum プロパティは、ノブに表示される値の範囲を指定します。この範囲は、Minimum および Maximum プロパティ(**double** 型)を使用して指定されます。C1Knob コントロールのデフォルトの範囲は0～ 100 です。

Value プロパティは、ゲージの現在の値を指定します。C1Knob コントロールでは、C1GaugePointer 要素がこの値を指し示すことによってこれが視覚的に示されます。C1Knob コントロールのデフォルトの Value は 50 です。

C1Knob の StartAngle および SweepAngle

範囲を定義したら、Minimum および Maximum 値に対応する角度を指定できます。StartAngle は、Value プロパティが範囲の Minimum 値に設定されているときのポインタの位置を定義します。SweepAngle は、Value プロパティが範囲の Maximum 値に設定されているときの StartAngle からの回転角度を指定します。

すべての角度は、コントロールの上中央から時計回りに度単位で指定されます。角度には負の値も指定できますが、SweepAngle の絶対値が 360 度を超えることはできません。

C1Knob の操作

InteractionMode プロパティは、実行時にコントロールで使用できる操作を制御します。つまり、ユーザーがノブを動かすために、クリック、ドラッグ、またはその両方を使用できるかどうかを選択できます。

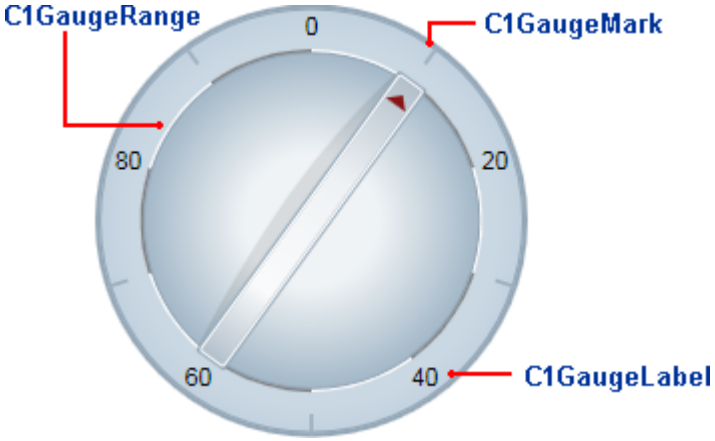
InteractionMode プロパティには、次の KnobInteractionMode 値の1つを設定できます。

値	説明
Drag	ユーザーがドラッグすることでポインタが移動します。
Click	ユーザーがノブ内をクリックすることでポインタが移動します。
ClickOrDrag	ユーザーがノブ内をクリックするか、ポインタをドラッグすることでポインタが移動します。

デフォルトでは、InteractionMode プロパティは **Click** に設定されています。

C1Knob のデコレータ

C1Knob コントロールには、デフォルトでは青灰色の背景とポインタのみが表示されます。C1RadialGauge や C1LinearGauge と同様に、**C1GaugeMark**、**C1GaugeLabel**、および **C1GaugeRange** 要素をゲージの Decorators コレクションに追加することで、グリッドのインジケータをカスタマイズできます。



これらのデコレータはスケール上の特定の位置に表示され、その位置は From、To、および Interval プロパティの値で決定されます。

上の画像では、1つの C1GaugeMark 要素と1つの C1GaugeLabel 要素が表示されています。

```
<!-- 目盛りマークの追加 -->
<Gauge:C1GaugeMark Interval="20" Alignment="In" From="10" />

<!-- ラベルマークの追加 -->
<Gauge:C1GaugeLabel Interval="20" Alignment="Center" Location="0.9" To="80" />
```

C1GaugeLabel 要素は、スケールに沿って 10 から 90 までの値のラベルを表示します。この C1GaugeMark 要素は、20 きざみに目盛りマークを表示します。

スケールを表示するほかに、いくつかの C1GaugeRange 要素をゲージの Decorators コレクションに追加することで、スケールの一部を強調表示できます。

上の画像では、10 個の C1GaugeRange 要素が表示されています。

```
<!-- 10 個の色付き範囲の追加 -->
<Gauge:C1GaugeRange From="0" To="10" Location="0.7" Fill="White" />
<Gauge:C1GaugeRange From="10" To="20" Location="0.7" Fill="Gray" />
<Gauge:C1GaugeRange From="20" To="30" Location="0.7" Fill="White" />
<Gauge:C1GaugeRange From="30" To="40" Location="0.7" Fill="Gray" />
<Gauge:C1GaugeRange From="40" To="50" Location="0.7" Fill="White" />
<Gauge:C1GaugeRange From="50" To="60" Location="0.7" Fill="Gray" />
<Gauge:C1GaugeRange From="60" To="70" Location="0.7" Fill="White" />
<Gauge:C1GaugeRange From="70" To="80" Location="0.7" Fill="Gray" />
<Gauge:C1GaugeRange From="80" To="90" Location="0.7" Fill="White" />
<Gauge:C1GaugeRange From="90" To="100" Location="0.7" Fill="Gray" />
```

これらの C1GaugeRange 要素は、白色および灰色の範囲を示します。各 C1GaugeRange 要素は、スケールに沿った1つの曲線型の帯として表示されます。帯の色は Fill プロパティによって決定され、位置は From および To プロパティによって決定されます。帯の太さは、StrokeThickness プロパティを使用して制御できます。

C1Knob デコレータの場所

各デコレータ要素には、要素が表示される位置を指定する Location プロパティがあります。このプロパティには、0（ゲージの中心）から1（ゲージの外縁）までを指定できます。ゲージコントロールには、すべてのデコレータの配置を制御する Radius プロパティもあり、これも0から1までを指定できます。半径プロパティのデフォルト値は 0.8 です。この場合は、すべてのデコレータがコントロール内に表示されます。

たとえば、C1GaugeLabel の Location プロパティを 1.1 に設定すると、ラベルはノブの外側にオフセットされて表示されます。Radius プロパティは 0.8 に設定されているため、ラベルはコントロール内に描画されます（この場合、ラベルの実際の位置は、1.1 * 0.8 = 0.88 と計算できます）。

タスク別ヘルプ

タスクベースのヘルプは、ユーザーの皆様が Visual Studio のプログラミングに精通しており、ゲージコントロールを使用する一般的な方法を理解していることを前提としています。**Gauges for UWP** 製品を初めて使用される場合は、まず「**クイックスタート**」を参照してください。

このセクションの各トピックは、**Gauges for UWP** 製品を使用して特定のタスクを実行するための方法を提供します。

また、タスクベースのヘルプトピックは、新しい UWP プロジェクトが作成されており、そのプロジェクトにゲージコントロールが追加されていることを前提としています。

開始値の設定

このトピックでは、C1LinearGauge コントロールの Value プロパティを変更します。Value プロパティは、現在選択されている値を決定します。デフォルトでは、C1LinearGauge コントロールは、最初に Value が**0**に設定されますが、設計時、XAML、またはコードでこの値をカスタマイズできます。このトピックでは、C1LinearGauge コントロールの Value を設定しますが、同じ手順を使用して、他のコントロールの Value をカスタマイズすることもできます。

設計時

実行時に C1LinearGauge コントロールの Value プロパティを設定するには、次の手順に従います。

1. C1LinearGauge コントロールをクリックして選択します。
2. [プロパティ]ウィンドウに移動し、Value プロパティの横にあるテキストボックスに値（たとえば、20）を入力します。
   これで、Value プロパティは指定された値に設定されます。

XAML の場合

たとえば、Value プロパティを設定するには、`Value="20"` を `<Gauge:C1LinearGauge>` タグに追加します。次のようになります。

XAMLマークアップ
<Gauge:C1LinearGauge Height="89" Margin="47,57,33,43" Name="C1LinearGauge1" Width="287" Value="20">

コードの場合

たとえば、Value プロパティを設定するには、プロジェクトに次のコードを追加します。

▶ Visual Basic コードの書き方

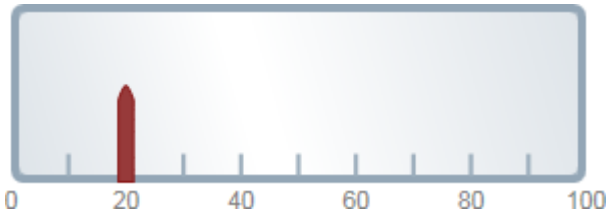
Visual Basic
C1LinearGauge1.Value = 20

▶ C# コードの書き方

C#
C1LinearGauge1.Value = 20;

プロジェクトの実行と確認

最初に、ゲージの C1GaugePointer は、指定された Value に設定されます。



最小値および最大値の設定

Minimum および Maximum プロパティを使用して、ゲージの数値範囲を設定できます。Minimum および Maximum 値は、設計時に、XAML、またはコードでカスタマイズできます。このトピックでは、C1LinearGauge コントロールの Minimum および Maximum プロパティを設定しますが、同じ手順を使用して、他のコントロールの Minimum と Maximum をカスタマイズすることもできます。

> **メモ:**Minimum および Maximum プロパティを設定する場合は、Minimum を Maximum より小さくする必要があります。また、Value プロパティは、Minimum から Maximum までの範囲内の値に設定してください(デフォルトは0です。これは、下で設定する範囲に入っています)。

設計時

実行時に C1LinearGauge の Minimum および Maximum を設定するには、次の手順に従います。

1. C1LinearGauge コントロールをクリックして選択します。
2. [プロパティ]ウィンドウに移動し、Maximum プロパティの横に値(たとえば、**50**)を入力します。
3. [プロパティ]ウィンドウで、Minimum の横に値(たとえば、**-50**)を入力します。
   これで、Minimum と Maximum の値が設定されます。

XAML の場合

XAML で C1LinearGauge コントロールの Minimum と Maximum を設定するには、`Maximum="50" Minimum="-50"` を `<Gauge:C1LinearGauge>` タグに追加します。次のようになります。

XAMLマークアップ
<Gauge:C1LinearGauge Height="89" Margin="47,57,33,43" Name="C1LinearGauge1" Width="287" Maximum="50" Minimum="-50">

コードの場合

C1LinearGauge コントロールの Minimum と Maximum を設定するには、次のコードをプロジェクトに追加します。

▶ Visual Basic コードの書き方

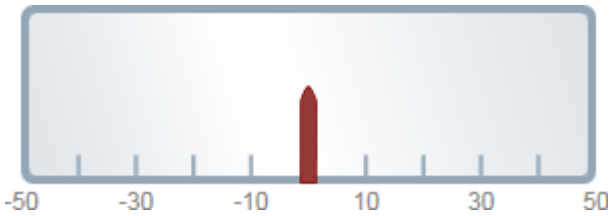
Visual basic
C1LinearGauge1.Minimum = -50 C1LinearGauge1.Maximum = 50

▶ C# コードの書き方

C#
C1LinearGauge1.Minimum = -50; C1LinearGauge1.Maximum = 50;

プロジェクトの実行と確認

実行時に、選択された範囲にゲージが制限されます。



ゲージへのラベルの追加

[プロパティ]ウィンドウ、XAML、またはコードで、**C1RadialGauge** コントロールのラベルを追加およびカスタマイズできます。このトピックでは、C1RadialGauge コントロールの **C1GaugeLabel** プロパティを設定しますが、同じ手順を使用して、他のコントロールの C1GaugeLabel をカスタマイズすることもできます。

設計時

設計時に[プロパティ]ウィンドウで C1RadialGauge コントロールにラベルを追加するには、次の手順に従います。

1. C1RadialGauge コントロールをクリックして選択します。
2. [プロパティ]ウィンドウに移動し、[**デコレータ**]項目の横にある**省略符**ボタンをクリックします。[**デコレータ**]コレクションエディタが開きます。
3. エディタの左上にあるドロップダウンリストで C1GaugeLabel を選択し、[**追加**]ボタンをクリックします。C1GaugeLabel デコレータがコレクションに追加されて選択されます。
4. 右側の[プロパティ]ペインで、C1GaugeLabel 要素の Location を **1** に設定します。
5. C1GaugeLabel 要素の Interval を **20** に設定します。
   これで、コントロールのラベルが設定されます。

XAML の場合

XAML で C1RadialGauge コントロールにラベルを追加するには、`<Gauge:C1GaugeLabel>` タグを `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

XAMLマークアップ

```
<Gauge:C1RadialGauge Height="189" Margin="42,29,188,31" Name="C1RadialGauge1" Width="189">
    <Gauge:C1GaugeLabel Interval="20" Location="1" />
 </Gauge:C1RadialGauge>
```

コードの場合

ウィンドウを右クリックし、[**コードの表示**]を選択してコードエディタを開きます。コードをメインクラスに追加します。次のようになります。

▶ Visual Basic コードの書き方

Visual Basic

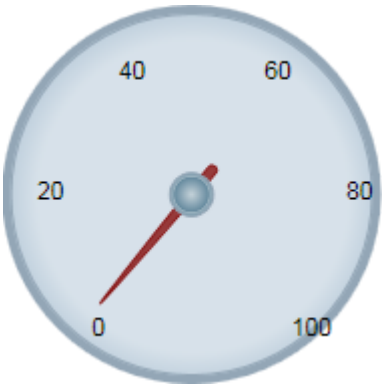
```
Public Sub New()
  InitializeComponent()
    Dim c1gl As New C1.Xaml.Gauge.C1GaugeLabel
    c1gl.Location = 1
    c1gl.Interval = 20
    Me.C1RadialGauge1.Decorators.Add(c1gl)
  End Sub
```

C# コードの書き方

```
C#
public MainPage(){
  InitializeComponent();
     C1.Xaml.Gauge.C1GaugeLabel c1gl = new C1.Xaml.Gauge.C1GaugeLabel();
     c1gl.Location = 1;
     c1gl.Interval = 20;
     this.C1RadialGauge1.Decorators.Add(c1gl);
  }
```

プロジェクトの実行と確認

C1RadialGauge コントロールにラベルが表示されます。



ゲージへの目盛りマークの追加

[プロパティ]ウィンドウ、XAML、またはコードで、**C1LinearGauge** コントロールに目盛りマークを追加できます。このトピックでは、C1LinearGauge コントロールの **C1GaugeMark** プロパティを設定しますが、同じ手順を使用して、他のコントロールの C1GaugeMark をカスタマイズすることもできます。

設計時

設計時に[プロパティ]ウィンドウで C1LinearGauge コントロールに目盛りマークを追加するには、次の手順に従います。

1. C1LinearGauge コントロールをクリックして選択します。
2. [プロパティ]ウィンドウに移動し、[**デコレータ**]項目の横にある**省略符**ボタンをクリックします。[**デコレータ**]コレクションエディタが開きます。
3. エディタの左上にあるドロップダウンリストで C1GaugeMark を選択し、[**追加**]ボタンをクリックします。C1GaugeMark デコレータがコレクションに追加されて選択されます。
4. 右側の[プロパティ]ペインで、C1GaugeMark 要素の Location を **1.1** に設定します。
5. C1GaugeLabel 要素の Interval を **20** に設定します。
6. エディタの左上にあるドロップダウンリストで C1GaugeMark を選択し、[**追加**]ボタンをクリックします。2番目の C1GaugeMark デコレータがコレクションに追加されて選択されます。
7. 右側の[プロパティ]ペインで、C1GaugeMark 要素の Location を **1.1** に設定します。
8. C1GaugeLabel 要素の Interval を **10** に設定します。
9. エディタの左上にあるドロップダウンリストで C1GaugeMark を選択し、[**追加**]ボタンをクリックします。3番目の C1GaugeMark デコレータがコレクションに追加されて選択されます。
10. 右側の[プロパティ]ペインで、C1GaugeMark 要素の Location を **1.1** に設定します。
11. C1GaugeLabel 要素の Interval を 5 に設定します。

XAML の場合

XAML で C1LinearGauge コントロールにラベルを追加するには、3つの <Gauge:C1GaugeMark> タグを <Gauge:C1LinearGauge> タグに追加します。

次のようになります。

XAMLマークアップ

```
<Gauge:C1LinearGauge Height="89" Margin="90,72,41,88" Name="C1LinearGauge1" Width="287">
    <Gauge:C1GaugeMark Interval="20" Location="1.1" />
    <Gauge:C1GaugeMark Interval="10" Location="1.1" />
    <Gauge:C1GaugeMark Interval="5" Location="1.1" />
 </Gauge:C1LinearGauge>
```

コードの場合

ウィンドウを右クリックし、[**コードの表示**]を選択してコードエディタを開きます。コードをメインクラスに追加します。次のようになります。

▶ Visual Basic コードの書き方

Visual Basic

```
Public Sub New()
 InitializeComponent()
    Dim c1gm1 As New C1.Xaml.Gauge.C1GaugeMark
    c1gm1.Location = 1.1
    c1gm1.Interval = 20
    Me.C1LinearGauge1.Decorators.Add(c1gm1)
    Dim c1gm2 As New C1.Xaml.Gauge.C1GaugeMark
    c1gm2.Location = 1.1
    c1gm2.Interval = 10
    Me.C1LinearGauge1.Decorators.Add(c1gm2)
    Dim c1gm3 As New C1.Xaml.Gauge.C1GaugeMark
    c1gm3.Location = 1.1
    c1gm3.Interval = 5
    Me.C1LinearGauge1.Decorators.Add(c1gm3)
 End Sub
```

▶ C# コードの書き方

C#

```
public MainPage(){
 InitializeComponent();
    C1.Xaml.Gauge.C1GaugeLabel c1gm1 = new C1.Xaml.Gauge.C1GaugeMark();
    c1gm1.Location = 1.1;
    c1gm1.Interval = 20;
    this.C1LinearGauge1.Decorators.Add(c1gm1);
    C1.Xaml.Gauge.C1GaugeLabel c1gm2 = new C1.Xaml.Gauge.C1GaugeMark();
    c1gm2.Location = 1.1;
   c1gm2.Interval = 10;
    this.C1LinearGauge1.Decorators.Add(c1gm2);
    C1.Xaml.Gauge.C1GaugeLabel c1gm3 = new C1.Xaml.Gauge.C1GaugeMark();
    c1gm3.Location = 1.1;
    c1gm3.Interval = 5;
    this.C1LinearGauge1.Decorators.Add(c1gm3);
```

```
}
```

プロジェクトの実行と確認

C1LinearGauge コントロールに3種類のサイズの目盛りマークが表示されます。

目盛りマークのカスタマイズ

デフォルトでは、ゲージの目盛りマークは青灰色の四角形として描画されます。C1GaugeMark 要素の **Template** プロパティにカスタムテンプレートを割り当てることで、目盛りマークの外観をカスタマイズできます。以下の手順では、**C1GaugeMark** の外観を定義する新しい **DataTemplate** を作成した後、そのテンプレートを **C1RadialGauge** コントロールの C1GaugeMark 要素の Template プロパティに割り当てます。

次の手順に従います。

1. XAML ビューに切り替えて、3つの `<Gauge:C1GaugeMark>` タグを `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

   XAMLマークアップ

   ```
   <Gauge:C1RadialGauge Height="89" Margin="90,72,41,88" Name="C1RadialGauge1" Width="287">
       <Gauge:C1GaugeMark From="0" To="100" Interval="10" />
       <Gauge:C1GaugeMark Interval="5" Location="1.1" />
   </Gauge:C1RadialGauge>
   ```

2. UserControl タグの真下に次のマークアップを追加して、テンプレートを追加します。

   XAMLマークアップ

   ```
   <UserControl.Resources>
       <!-- ゲージマークの描画に使用されるテンプレート -->
       <DataTemplate x:Key="MyMarkTemplate">
           <Rectangle Width="4" Height="18" Fill="BlueViolet" Stroke="Black" StrokeThickness=".5"/>
       </DataTemplate>
   </UserControl.Resources>
   ```

   このテンプレートは、目盛りマークの外観を定義します。

3. 次に、最初に追加した C1GaugeMark 要素に Template プロパティを設定して、新しいテンプレートのキーを参照します。

   XAMLマークアップ

   ```
   <Gauge:C1GaugeMark From="0" To="100" Interval="10" Template="{StaticResource MyMarkTemplate}"/>
   ```

プロジェクトの実行と確認

C1RadialGauge コントロールに、カスタマイズされた目盛りマークが表示されます。

マークは、Location プロパティで指定された位置から内側に向かって描画されています。マークの表示に使用される四角形の **Height** を増やすと、目盛りマークはゲージの中心に向かって長くなります。マークを外側に向けて延ばすには、C1GaugeMark 要素の Location プロパティを変更します。また、目盛りマークの表示に使用される要素はスケールに沿って向きが変わりますが、ラベルの表示に使用される要素は向きが変わらないことがわかります(「**ゲージへのラベルの追加**」を参照)。

ゲージ形状のカスタマイズ

ほとんどの円形ゲージは丸形ですが、他の形状でゲージを作成することもできます。**C1RadialGauge** の形状をカスタマイズするには、次の手順に従う必要があります。

- ゲージの形状を選択します。
- ゲージの形状を考慮し、ポインタの位置に合わせて **PointerOrigin** プロパティを設定します。
- ゲージの **Background** プロパティを **Transparent** に、**BorderThickness** プロパティを 0 に設定して、デフォルトの丸い背景を非表示にします。
- 要素を Face レイヤに追加して、新しいゲージの形状を表示します。

上の手順に基づいてカスタマイズされた形状の C1RadialGauge を作成するには、次の手順に従います。

1. XAML ビューに切り替えて、`<Gauge:C1RadialGauge>` タグを変更します。次のようになります。

   XAMLマークアップ

   ```
   <Gauge:C1RadialGauge Height="189" Margin="102,34,127,26" Name="C1RadialGauge1" Width="189"
   StartAngle="-160" SweepAngle = "140">
   </Gauge:C1RadialGauge>
   ```

   これで、C1RadialGauge コントロールの初期プロパティが設定されます。

2. XAML ビューで、`PointerOrigin="0.8,0.5"` を `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

   XAMLマークアップ

   ```
   <Gauge:C1RadialGauge Height="189" Margin="102,34,127,26" Name="C1RadialGauge1" Width="189"
   StartAngle="-160" SweepAngle = "140" PointerOrigin="0.8,0.5">
   </Gauge:C1RadialGauge>
   ```

   PointerOrigin プロパティは、C1RadialGauge コントロールの C1GaugePointer の開始位置を設定します。

3. XAML ビューで、`Background="Transparent"` を `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

   XAMLマークアップ

   ```
   <Gauge:C1RadialGauge Height="189" Margin="102,34,127,26" Name="C1RadialGauge1" Width="189"
   StartAngle="-160" SweepAngle = "140" PointerOrigin="0.8,0.5" Background="Transparent">
   ```

```
</Gauge:C1RadialGauge>
```

これで、C1RadialGauge コントロールが透明に表示されます。

4. XAML ビューで、`BorderThickness="0"` を `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

XAMLマークアップ

```
<Gauge:C1RadialGauge Height="189" Margin="102,34,127,26" Name="C1RadialGauge1" Width="189"
StartAngle="-160" SweepAngle = "140" PointerOrigin="0.8,0.5" Background="Transparent"
BorderThickness="0">
</Gauge:C1RadialGauge>
```

これで、C1RadialGauge コントロールが境界線なしで表示されます。

5. XAML ビューで、マークアップを `<Gauge:C1RadialGauge>` タグの後に追加します。次のようになります。

XAMLマークアップ

```
<Gauge:C1RadialGauge Height="189" Margin="102,34,127,26" Name="C1RadialGauge1" Width="189"
StartAngle="-160" SweepAngle = "140" PointerOrigin="0.8,0.5" Background="Transparent"
BorderThickness="0">
   <!-- ゲージに目盛りの追加 -->
   <Gauge:C1GaugeMark Interval="10" Location="1"/>
   <Gauge:C1GaugeMark Interval="5" Location="1" />
</Gauge:C1RadialGauge>
```

これで、ゲージに C1GaugeMark 要素と目盛りマークが追加されます。

6. XAML ビューで、マークアップを `<Gauge:C1RadialGauge>` タグの後に追加します。次のようになります。

XAMLマークアップ

```
<Gauge:C1RadialGauge Height="189" Margin="102,34,127,26" Name="C1RadialGauge1" Width="189"
StartAngle="-160" SweepAngle = "140" PointerOrigin="0.8,0.5" Background="Transparent"
BorderThickness="0">
   <!-- ゲージに目盛りの追加 -->
   <Gauge:C1GaugeMark Interval="10" Location="1"/>
   <Gauge:C1GaugeMark Interval="5" Location="1" />
   <!-- カスタムシェープを含むフェースの追加 -->
   <Gauge:C1RadialGauge.Face>
      <Grid>
         <Grid.ColumnDefinitions>
            <ColumnDefinition Width="4*" />
            <ColumnDefinition Width="10*" />
            <ColumnDefinition Width="1*" />
         </Grid.ColumnDefinitions>
         <Border Grid.Column="1" Background="Black" BorderBrush="LightGray" BorderThickness="4"
CornerRadius="140,60,60,140"/>
      </Grid>
   </Gauge:C1RadialGauge.Face>
</Gauge:C1RadialGauge>
```

これで、ゲージにカスタマイズされた Face が追加されます。

プロジェクトの実行と確認

C1RadialGauge コントロールに、カスタマイズされたフェイスが表示されます。

C1RadialGauge コントロールの Face は、さまざまにカスタマイズできます。たとえば、**UWP Edition** と共にインストールされる **GaugeSamples** サンプルの **SpeedometersPage.xaml** ページには、次のカスタマイズされたゲージがあります。

ポインタの外観のカスタマイズ

デフォルトでは、**C1GaugePointer** は先細の茶色の四角形として表示され、ポインタキャップはグラデーション付きの灰色の円として表示されます。どちらの外観もカスタマイズできます。以下の手順では、**C1RadialGauge** コントロールの C1GaugePointer と **PointerCap** の外観をカスタマイズします。

次の手順に従います。

1. C1RadialGauge コントロールをクリックして選択します。
2. XAML ビューに切り替えて、`<Gauge:C1RadialGauge>` タグに `PointerFill="SkyBlue" PointerStroke="CornflowerBlue"` を追加します。次のようになります。

   XAMLマークアップ

   ```
   <Gauge:C1RadialGauge Height="226" Margin="22,24,0,12" Name="C1RadialGauge1" Width="256"
   PointerFill="SkyBlue" PointerStroke="CornflowerBlue">
    </Gauge:C1RadialGauge>
   ```

   これで、C1GaugePointer の色がカスタマイズされます。

3. XAML ビューで、`PointerCapStroke="CornflowerBlue"` を `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

   XAMLマークアップ

   ```
   <Gauge:C1RadialGauge Height="226" Margin="22,24,0,12" Name="C1RadialGauge1" Width="256"
   PointerFill="SkyBlue" PointerCapStroke="CornflowerBlue" PointerStroke="CornflowerBlue">
    </Gauge:C1RadialGauge>
   ```

   これで、PointerCap の輪郭の色がカスタマイズされます。

4. XAML ビューで、次の `<Gauge:C1RadialGauge.PointerCapFill>` マークアップを `<Gauge:C1RadialGauge>` タグに追加します。次のようになります。

XAMLマークアップ

```
<Gauge:C1RadialGauge Height="226" Margin="22,24,0,12" Name="C1RadialGauge1" Width="256"
PointerFill="SkyBlue" PointerCapStroke="CornflowerBlue" PointerStroke="CornflowerBlue" >
    <Gauge:C1RadialGauge.PointerCapFill>
        <LinearGradientBrush>
            <GradientStop Color="CornflowerBlue" Offset="0"/>
            <GradientStop Color="SkyBlue" Offset="1"/>
        </LinearGradientBrush>
    </Gauge:C1RadialGauge.PointerCapFill>
 </Gauge:C1RadialGauge>
```

これで、C1RadialGauge コントロールの PointerCap に直線方向のグラデーションが追加されます。

プロジェクトの実行と確認

C1RadialGauge コントロールに、カスタマイズされた C1GaugePointer と PointerCap が表示されます。