

LinearGauge for ASP.NET Web Forms

2018.04.12 更新

グレースィティ株式会社

目次

製品の概要	2
ComponentOne for ASP.NET Web Forms のヘルプ	2
主な特長	3
クイックスタート	4
手順 1: ページへの C1LinearGauge の追加	4
手順 2: C1LinearGauge の外観のカスタマイズ	4-6
手順 3: コントロールの動作のカスタマイズ	6-7
LinearGauge for ASP.NET Web Forms の使い方	8
Gauge コントロールを使用する理由	8
C1LinearGauge の値	8-9
C1LinearGauge の向き	9
C1LinearGauge のインジケータ	9-10
C1LinearGauge ポインタ	10-11
C1LinearGauge の表面とカバー	11
タスク別ヘルプ	12
コードでの C1LinearGauge の作成	12-13
ラベルを追加する	13-14
目盛りを追加する	14-15
対数マーカーを追加する	15-16
範囲を追加する	16-17
コントロールのサイズ変更	17-18
開始値の設定	18-19
最小値と最大値の設定	19-20
アニメーションを設定する	20-21

製品の概要

LinearGauge for ASP.NET Web Forms を使用して、動的で独自の方法で情報を簡単に表示できます。**LinearGauge for ASP.NET Web Forms** には、**C1LinearGauge** という1つのコントロールが含まれています。

ComponentOne for ASP.NET Web Forms のヘルプ

ComponentOne for ASP.NET Web Forms の各コントロールで共通したトピック、アセンブリの追加、テーマの適用、クライアント側情報などについては「[ASP.NET Web Forms ユーザーガイド](#)」を参照してください。

主な特長

C1LinearGauge は、以下の独特な主要機能を備えています。

- **スタイル設定を簡素化**

すべてのゲージ要素はニーズに合わせてカスタマイズできます。ComponentOne のゲージコントロールのスタイル変更は、プロパティを単に変更するだけです。ゲージの主な要素はすべて表面に表示されています。カスタムの色、塗りつぶし、フォントなどを持つゲージを作成できます。

- **範囲**

色付きの範囲をゲージに追加し、特定の値範囲に注目させます。単純なプロパティを使用し、開始点と終了点や、位置、サイズ、および外観をカスタマイズします。開始と終了幅を指定して非線形範囲を作成し、成長を表示したり視覚的なアピール力を任意のゲージに追加したりできます。

- **目盛記号とラベル**

目盛記号とラベルを定義します。単純なプロパティを使用し、間隔、位置、および外観をカスタマイズします。書式をゲージラベルに適用します。たとえば、標準 .NET 書式文字列を使用し、ラベルを通貨やパーセント書式に書式設定します。

- **向き**

Orientation プロパティを使用し、線形ゲージをページ上の水平または垂直方向に配置できます。

- **豊富なカスタマイズ**

使用可能なスタイル属性の豊富なセットを使用し、ゲージの表面とカバーの形状をカスタマイズして希望する任意の外観を作成します。単純な形状を使用してガラス効果をシミュレートすることさえ可能です。形状が十分でない場合、画像を追加できます。回転、反転や、色相、彩度、輝度、不透明度の変更など、さまざまな効果を画像に適用できます。

- **スケールのカスタマイズ**

単純なプロパティを使用し、ゲージスケールの開始と終了角度を設定します。

- **ポイントのカスタマイズ**

多くの定義済みポイント形状から選択したり、形状をカスタマイズしたり、独自のカスタム画像をインポートしたりして、ポイントとして使用します。放射型と線形ゲージのポイント起点の正確な位置を指定します。これにより、ポイントをゲージの左右や下部に移動できます。

- **アニメーション**

ゲージでは簡単にスムーズなアニメーションを使用できます。アニメーションをカスタマイズすることも可能です。たとえば、ポイントアニメーションの場合、値が変化する際のアニメーションの時間間隔を設定できます。また、ソース値が頻繁に変化しすぎて確認できない場合は、ゲージコントロールの再描画の頻度を減らせます。

- **ライブデータ**

ゲージはデータが変化するとアニメーション表示されます。

- **対数マーカ**

Islogarithmic と **LogarithmicBase** プロパティによって、対数マーカを取得できます。

クイックスタート

このクイックスタートでは、**C1LinearGauge** コントロールの機能について学びます。**C1LinearGauge** は、独自のビューにデータを表示する機能を提供します。

手順 1: ページへの C1LinearGauge の追加

この手順では、Web サイトを作成して設定し、空の **C1LinearGauge** コントロールを追加します。**C1LinearGauge** コントロールを Web サイトに追加するには、以下の手順を実行します。

1. Visual Studio の[ファイル]メニューから、[新規作成]→[プロジェクト]の順に選択します。[新しいプロジェクト]ダイアログボックスが表示されます。
2. [新しいプロジェクト]ダイアログボックスの左ペインで、言語を展開し、[Web]を選択します。右ペインで、[ASP.NET 空の Web アプリケーション]を選択し、自分のアプリケーションの[名前]を入力して、<OK>をクリックします。新しいアプリケーションが作成されます。
3. ソリューションエクスプローラでプロジェクトを右クリックして、コンテキストメニューから[新しい項目の追加]を選択します。
4. [参照の追加]ダイアログボックスで、**C1.Web.Wijmo.Controls** アセンブリを探して選択し、<OK>をクリックします。参照が追加されます。
5. ソリューションエクスプローラでプロジェクトを右クリックして、コンテキストメニューから[新しい項目の追加]を選択します。
6. [新しい項目の追加]ダイアログボックスで、テンプレートのリストから[Web フォーム]を選択し、項目 Default.aspx に名前を付けて、<追加>をクリックします。新しいページが開きます。
7. ソースビューで、次のマークアップをページの上部に追加し、**C1LinearGauge** アセンブリを登録します。

ソースビュー

```
<%@ Register Assembly="C1.Web.Wijmo.Controls.4" Namespace="C1.Web.Wijmo.Controls.C1LinearGauge" TagPrefix="cc1" %>
```

追加したアセンブリに応じて、上記で C1.Web.Wijmo.Controls.4 を C1.Web.Wijmo.Controls.3 に置き換える必要がある場合があります。

8. マウスをページの最初の <div></div> タグの間に入れて、次のマークアップを追加し、**C1LinearGauge** コントロールをページに追加します。

ソースビュー

```
<cc1:C1LinearGauge ID="C1LinearGauge1" runat="server"> </cc1:C1LinearGauge>
```

9. アプリケーションを実行し、ページが次のような表示になったことを確認します。



この手順では、**C1LinearGauge** コントロールをフォームに追加しました。ここでは、コントロールは書式設定されていません。クイックスタートの次の手順では、コンテンツをコントロールに追加します。

手順 2: C1LinearGauge の外観のカスタマイズ

この手順では、**C1LinearGauge** コントロールの外観をカスタマイズします。以下の手順では、「手順1/3: ページへの C1LinearGauge の追加」トピックが終了しており、ページに **C1LinearGauge** コントロールが追加されていることを前提とします。

デザインビューの場合

LinearGauge for ASP.NET Web Forms

デザインビューで **C1LinearGauge** コントロールをカスタマイズするには、以下の手順を実行します。

- 以下の手順を実行して、コントロールの **Width** と **Height** を設定します。
 1. コントロールを右クリックし、**[プロパティ]** を選択します。
 2. プロパティウィンドウで、**Options** プロパティを展開します。
 3. コントロールの **Width** と **Height** に任意の値を設定します。
- 以下の手順を実行して、コントロールの初期 **Value** を設定します。
 1. コントロールを右クリックし、**[プロパティ]** を選択します。
 2. **Behavior** プロパティを展開します。
 3. 目的の **Value** を設定します。
- 以下の手順を実行して、コントロールの **Face** の外観を変更します。
 1. コントロールを右クリックし、**[プロパティ]** を選択します。
 2. **[プロパティ]** ウィンドウで、**Appearance** プロパティを展開します。
 3. **Face** プロパティを展開します。
 4. **FaceStyle** プロパティを展開します。
 5. **Fill** プロパティを展開します。
 6. **Color**、**ColorBegin**、**ColorEnd**、**Type** などのプロパティを任意に設定して、ゲージの外観を変更します。



メモ: **FaceStyle** プロパティの下で、他のいくつかのプロパティを自由に設定して、**C1LinearGauge** コントロールの外観をカスタマイズできます。

ソースビューの場合

ソースビューで **C1LinearGauge** コントロールをカスタマイズするには、以下の手順を実行します。

- コントロールの **Width** と **Height** を設定し、実行時のゲージの初期 Value を設定するには、`<cc1:C1LinearGauge>` `</cc1:C1LinearGauge>` タグを次のように編集します。

```
<cc1:C1LinearGauge ID="C1LinearGauge1" runat="server" Width="400" Height="100" Value="50">
```
- コントロールの **Face** の外観を変更するには、`<cc1:C1LinearGauge>` `</cc1:C1LinearGauge>` タグの間にマウスを置き、次のマークアップを追加します。

```
<Face>  
  <FaceStyle Fill-LinearGradientAngle="270" Fill-ColorBegin="pink" Fill-ColorEnd="purple  
    Fill-Type="LinearGradient" StrokeWidth="4"></FaceStyle>  
</Face>
```
- 必要に応じて、**LinearGradientAngle**、**ColorBegin**、**ColorEnd**、**Type**、および **StrokeWidth** を設定して、**C1LinearGauge** コントロールの外観をカスタマイズします。

コードの場合

Width、**Height**、**Max**、**Min** の各値を設定して **C1LinearGauge** コントロールの外観をカスタマイズするには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

Visual Basic

```
'C1LinearGauge の最小値を設定します。  
C1LinearGauge1.Min = 10  
'C1LinearGauge の最大値を設定します。  
C1LinearGauge1.Max = 90  
'C1LinearGauge の高さを設定します。  
C1LinearGauge1.Height = 100  
'C1LinearGauge の幅を設定します。  
C1LinearGauge1.Width = 800  
'C1LinearGauge の塗りつぶしの開始色を設定します。  
C1LinearGauge1.Face.FaceStyle.Fill.ColorBegin = System.Drawing.Color.Pink  
'C1LinearGauge の塗りつぶしの終了色を設定します。  
C1LinearGauge1.Face.FaceStyle.Fill.ColorEnd = System.Drawing.Color.Purple  
'C1LinearGauge のグラデーションのタイプを設定します。  
C1LinearGauge1.Face.FaceStyle.Fill.Type = C1.Web.Wijmo.Controls.C1Chart.ChartStyleFillType.LinearGradient
```

C# でコードを書く場合

C#

```
//C1LinearGauge の最小値を設定します。
C1LinearGauge1.Min = 10;

//C1LinearGauge の最大値を設定します。
C1LinearGauge1.Max = 90;

//C1LinearGauge の高さを設定します。
C1LinearGauge1.Height = 100

//C1LinearGauge の幅を設定します。
C1LinearGauge1.Width = 800;

//C1LinearGauge の塗りつぶしの開始色を設定します。
C1LinearGauge1.Face.FaceStyle.Fill.ColorBegin = System.Drawing.Color.Pink;

//C1LinearGauge の塗りつぶしの終了色を設定します。
C1LinearGauge1.Face.FaceStyle.Fill.ColorEnd = System.Drawing.Color.Purple;

//C1LinearGauge のグラデーションのタイプを設定します。
C1LinearGauge1.Face.FaceStyle.Fill.Type = C1.Web.Wijmo.Controls.C1Chart.ChartStyleFillType.LinearGradient;
```

ここまでの成果

アプリケーションを実行して、ゲージが次の図のように表示されることを確認します。



手順 3: コントロールの動作のカスタマイズ

この手順では、スライダコントロールを追加します。実行時にスライダのつまみボタンが移動すると、ゲージコントロールの値が変化します。次の手順では、「[手順 2: コントロールの外観のカスタマイズ](#)」トピックが完了していると想定しています。

1. ソースビューで、次のマークアップを</cc1:C1LinearGauge> タグのすぐ下に追加して、テキストを追加します。

ソースビュー

```
<p>スライダのつまみボタンをドラッグ&ドロップして、ゲージの値を変更する:</p>
```

2. ソースビューで、直前に追加した</p> タグのすぐ下に次のマークアップを追加して、スライダコントロールをページに追加します。

ソースビュー

```
<div id="slider" style="width: 400px"></div>
```

これで、ページにスライダコントロールが含まれました。

3. ソースビューで、直前に追加した</div> タグのすぐ下に次のマークアップを追加して、スライダコントロールを有効にします。

ソースビュー

```
<script type="text/javascript">
    $(document).ready(function () {
        $("#<%= C1Slider1.ClientID %>").c1slider({
            value: $("#<%= C1LinearGauge1.ClientID %>").c1lineargauge("option", "value"),
            change: function (event, ui) {
                $("#<%= C1LinearGauge1.ClientID %>").c1lineargauge("option", "value", ui.value);
            }
        });
    });
</script>
```

LinearGauge for ASP.NET Web Forms

```
});  
});  
</script>
```

これで、スライダの値が変化すると、ゲージの値も変化して新しい値を反映するようになりました。

4. アプリケーションを実行し、次のように、加えた変更がコントロールに表示されることを確認します。



スライダのつまみボタンをドラッグ&ドロップして、ゲージの値を変更する：

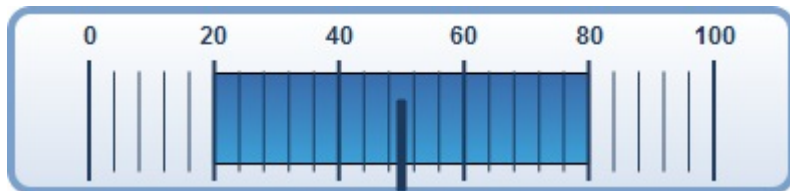


5. スライダのつまみボタンをクリックして、ドラッグ&ドロップ操作を実行します。**C1LinearGauge** コントロールの値が変わって、スライダコントロールの値を反映することに注意してください。

この手順では、コントロールの動作をカスタマイズしました。おめでとうございます、これでクイックスタートは終了です。

LinearGauge for ASP.NET Web Forms の使い方

C1LinearGauge は、線形のポインタを使用して、線形スケールに沿った値を表示します。値は **Value** プロパティによって表され、範囲は **Min** と **Max** プロパティで定義されます。**C1LinearGauge** コントロールは、標準的な温度計に類似した表示になります。



C1LinearGauge コントロールの作成と使用には通常、以下の手順が含まれます。

1. **C1LinearGauge** コントロールを作成して、メインプロパティである **Min**、**Max**、および **Orientation** を設定します。
2. スケールを表示するには **TickMajor** と **TickMinor** 装飾子を追加し、ラベルをカスタマイズするには **GaugeLabel** を追加します。
3. オプションとして、スケールの部分をハイライトさせるには **GaugeRange** 装飾子を追加します。範囲は通常、「低すぎる」、「正常」、「高すぎる」の各範囲を示すために使用されます。また、範囲を動的にして、**Value** プロパティが変更されたときに自動的に移動させることもできます。
4. オプションとして、テンプレートを用いてゲージ要素をカスタマイズします。
5. 表示したい値を表示させるには、**Value** プロパティを設定します。

Gauge コントロールを使用する理由

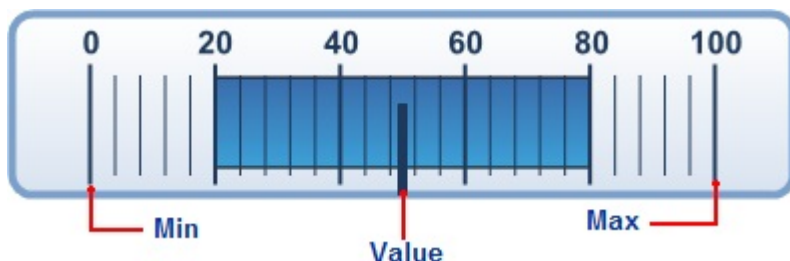
ゲージコントロールを使用する必要があるのか疑問に思うかもしれません。ゲージは1つの値を表示するだけであり、その値はゲージの代わりに単純なラベルを使用しても表示できます。

ゲージがより優れているのは、範囲も表示する点です。これによって、ユーザーは現在の値が低い、高い、または適切かどうかを瞬時に決定できます。確かに、2つの追加のラベルを使用すれば、範囲も現在の値も表示できますが、それによってユーザーインターフェースはさらに分かりにくいものになります。このため、多くのアプリケーションでは、進捗を単にラベルとして表示する代わりに、単純な線形ゲージである進捗インジケータが使用されています。

また、ゲージには、単純なラベル（あるいはスライダーやスクロールバー）よりも視覚的な魅力があり、アプリケーションの付加価値が高まります。

C1LinearGauge の値

C1LinearGauge コントロールの **Min**、**Max**、および **Value** プロパティを使用して、使用可能な範囲とその範囲内で選択された値を指定できます。



Min と **Max** プロパティは、ゲージが表示するように設計されている値の範囲を指定します。たとえば、温度計は $-40 \sim 100$ 度のスケール範囲、速度計は $0 \sim 140$ マイル/時の範囲に設定できます。範囲は、**Min** と **Max** プロパティ (**double** 型) から指定されます。**C1LinearGauge** コントロールのデフォルト範囲は $0 \sim 100$ です。

Value プロパティは、ゲージの現在の値を示します。**C1LinearGauge** コントロールでは、これは **Pointer** 要素がポイントしている値によって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**

LinearGauge for ASP.NET Web Forms

は 50 に設定されました。

C1LinearGauge の向き

C1LinearGauge コントロールは、**Orientation** プロパティを持っており、それを使用して、垂直または水平ゲージを作成できます。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水平方向に表示されます。



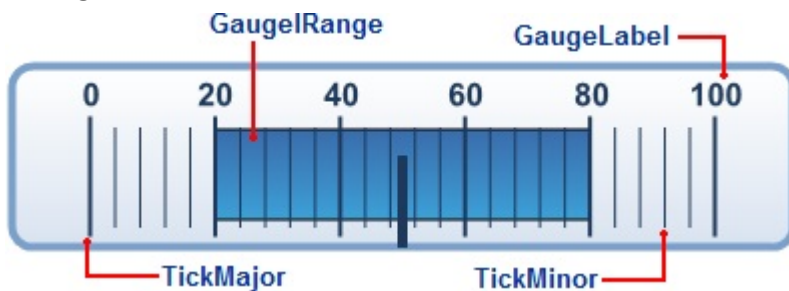
垂直ゲージを作成するには、**Orientation** プロパティを **Vertical** に設定します。



たとえば、垂直線形ゲージは温度計として使用できます。

C1LinearGauge のインジケータ

デフォルトでは、**C1LinearGauge** コントロールは単純な水平線形ゲージのみを表示します。また、大部分のアプリケーションでは、カスタムラベルと目盛記号で構成されるスケールを表示して、ユーザーが現在の値とその値のゲージ範囲内の位置を読み取れるようにしたい場合があります。これを実行するには、**TickMajor**、**TickMinor**、**GaugeRange**、および **GaugeLabel** 要素を追加します。



上の図では、カスタマイズされた **TickMajor** と **TickMinor** 要素が表示されます。

ソースビュー

```
<!-- 目盛り -->
```

```
<TickMajor Position="Inside" Offset="-11" Interval="20" Factor="12">
  <TickStyle Width="2">
    <Fill Color="#1E395B"> </Fill>
  </TickStyle>
</TickMajor>
<TickMinor Position="Inside" Visible="true" Offset="-11" Interval="4" Factor="10">
  <TickStyle Width="1">
    <Fill Color="#1E395B"> </Fill>
  </TickStyle>
</TickMinor>
```

GaugeLabel も次のように設定されています。

ソースビュー

```
<!-- ラベルを追加します。-->
<Labels>
  <LabelStyle FontSize="12pt" FontWeight="800">
    <Fill Color="#1E395B"> </Fill>
  </LabelStyle>
</Labels>
```

スケールの表示に加えて、スケール範囲の部分をハイライトしたい場合があります。たとえば、赤いマーカを追加して、その範囲の値が低すぎる(売上)か高すぎる(費用)ことを示したい場合があります。これを実行するには、1つ以上の **GaugeRange** 要素を追加します。

上の図では、色付きの **GaugeRange** 要素が表示されます。

ソースビュー

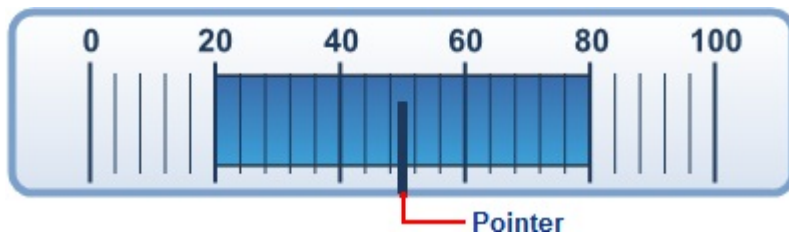
```
<!-- 色付きの範囲を追加します。-->
<Ranges>
<cc1:GaugeRange StartValue="20" EndValue="80" StartDistance="0.85" EndDistance="0.85" StartWidth="0.5"
EndWidth="0.5">
  <RangeStyle>
    <Fill LinearGradientAngle="90" ColorBegin="#3DA1D8" ColorEnd="#3A6CAC" Type="LinearGradient"> </Fill>
  </RangeStyle>
</cc1:GaugeRange>
</Ranges>
```

GaugeRange 要素は、スケールに沿った色付きの帯域を表示します。帯域の色は **RangeStyle** プロパティによって決定され、位置は **StartValue** と **EndValue** プロパティによって決定されます。範囲の幅を制御するには、**StartWidth** と **EndWidth** プロパティを使用します。

C1LinearGauge ポインタ

C1LinearGauge には、コントロールの選択された **Value** を示すポインタが含まれます。ポインタは **Pointer** 要素で構成されます。

LinearGauge for ASP.NET Web Forms



デフォルトでは **Pointer** 要素は青いテーパー要素として表示されます。**Pointer** 要素の外観をカスタマイズするには、**GaugePointer** クラスのいくつかのプロパティ(**Length**、**Offset**、**Shape**、**PointerStyle**、**Template**、**Visible**、および**Width** プロパティを含む)を設定します。

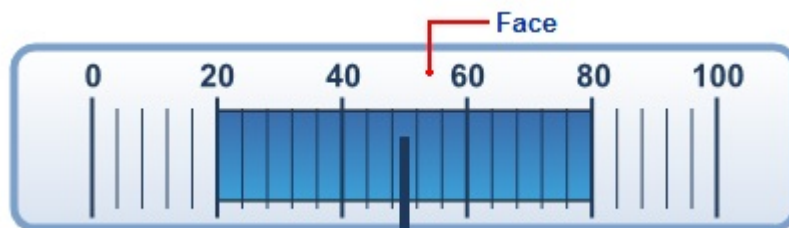
上の図では、**Pointer** 要素がカスタマイズされています。

ソースビュー

```
<!-- ポインタ -->
<Pointer Length="0.5" Shape="Rect">
  <PointerStyle Stroke="#1E395B">
    <Fill Color="#1E395B"></Fill>
  </PointerStyle>
</Pointer>
```

C1LinearGauge の表面とカバー

時計と同様に、**C1LinearGauge** コントロールには、**Face** 要素が含まれます。**Face** は背景の上に表示されますが、ポインタやその他の要素の背後に表示されます。たとえば、下の図では、**Face** はゲージの要素の背後に表示される単純な灰色の円です。



上の図では、**Face** 要素はカスタマイズされています。

ソースビュー

```
<!-- 表面 -->
<Face>
  <FaceStyle Stroke="#7BA0CC" StrokeWidth="4">
    <Fill LinearGradientAngle="270" ColorBegin="#FFFFFF" ColorEnd="#D9E3F0" Type="LinearGradient"></Fill>
  </FaceStyle>
</Face>
```

デフォルトでは、**Face** 要素は青灰色の円として表示されます。**Face** 要素の外観をカスタマイズするには、**GaugeFace** クラスのいくつかのプロパティ (**FaceStyle** プロパティを含む)を設定します。

タスク別ヘルプ

タスク別ヘルプは、ASP.NET のプログラミングに精通し、コントロールの一般的な使用方法を理解しているユーザーを対象としています。ヘルプに記述された手順に従うことによって、**C1LinearGauge**のさまざまな機能をデモンストレーションするプロジェクトを作成して、**C1LinearGauge** の機能を理解できます。

各トピックでは、**C1LinearGauge** コントロールを使用した特定のタスクのソリューションを示します。タスク別ヘルプの各トピックでは、新しい ASP.NET プロジェクトを作成し、必要なアセンブリへの参照を追加済みであることも前提となります。

コードでの C1LinearGauge の作成

コードで **C1LinearGauge** コントロールを作成するのは、とても簡単です。以下の手順では、**PlaceHolder** コントロールをページに追加して、インポートステートメントを追加した後、**C1LinearGauge** を追加してカスタマイズしてから、このコントロールを **PlaceHolder** に追加します。

以下の手順を実行します。

1. デザインビューで、ページをダブルクリックして **Page_Load** イベントを作成し、コードビューに切り替えます。
2. 次のステートメントをコードエディタの上部に追加し、必要な名前空間をインポートします。

Visual Basic コードの書き方

```
Visual Basic
Imports C1.Web.Wijmo.Controls.C1Gauge
```

C#コードの書き方

```
C#
using C1.Web.Wijmo.Controls.C1Gauge;
```

3. 次のコードを **Page_Load** イベントに追加し、**C1LinearGauge** コントロールを作成してカスタマイズします。

Visual Basicコードの書き方

```
Visual Basic
'新しい C1LinearGauge を作成します。
Dim C1LG As New C1LinearGauge
' コントロールのサイズと値を設定します。
C1LG.Height = 100
C1LG.Width = 500
C1LG.Min = 0
C1LG.Max = 100
C1LG.Value = 60
'C1LinearGauge を Placeholder コントロールに追加します。
Dim Placeholder1 As New Placeholder
Placeholder1.Controls.Add(C1LG)
form1.Controls.Add(Placeholder1)
```

C#コードの書き方

```
C#
// 新しい C1LinearGauge を作成します。
C1LinearGauge C1LG = new C1LinearGauge();
```

LinearGauge for ASP.NET Web Forms


```
//コントロールのサイズと値を設定します。
C1LG.Height = 100;
C1LG.Width = 500;
C1LG.Min = 0;
C1LG.Max = 100;
C1LG.Value = 60;
//C1LinearGauge を Placeholder コントロールに追加します。
Placeholder Placeholder1 = new Placeholder();
Placeholder1.Controls.Add(C1LG);
form1.Controls.Add(Placeholder1);
```

✔このトピックの作業結果

アプリケーションを実行し、**C1LinearGauge** コントロールが作成されたことを確認します。

ラベルを追加する

この手順では、**C1LinearGauge** コントロールにラベルを追加します。以下の手順では、ページに **C1LinearGauge** コントロールが既に追加されていることを前提とします。

 **メモ:** **C1LinearGauge** コントロールにはデフォルトでラベルが追加されます。ラベルが表示されない場合は、**Visible** プロパティを True に設定します。

デザインビューの場合

デザインビューで **C1LinearGauge** コントロールにラベルを追加するには、以下の手順を実行します。

1. コントロールを右クリックし、**[プロパティ]**を選択します。
2. プロパティウィンドウで、**Behavior** プロパティを展開します。
3. **Labels** プロパティを展開します。
4. **Visible** を True に設定します。

ソースビューの場合

C1LinearGauge コントロールにラベルを追加するには、
<cc1:C1LinearGauge> </cc1:C1LinearGauge>
タグの間にマウスを置き、次のマークアップを追加します。
<Labels Visible="true"></Labels>

Visible を true に設定します。

コードの場合

C1LinearGauge コントロールにラベルを追加するには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

Visual Basic

```
' C1LinearGauge にラベルを追加します
C1LinearGauge1.Labels.Visible = True
```

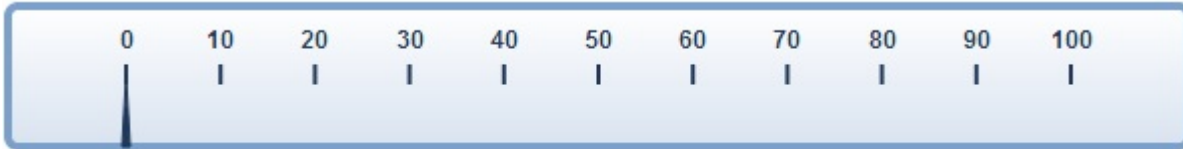
C# でコードを書く場合

C#

```
// C1LinearGauge にラベルを追加します  
C1LinearGauge1.Labels.Visible = true;
```

ここまでの成果

アプリケーションを実行して、ゲージが次の図のようにラベル付きで表示されることを確認します。



目盛りを追加する

この手順では、**C1LinearGauge** に目盛りを追加します。以下の手順では、ページに **C1LinearGauge** コントロールが既に追加されていることを前提とします。

デザインビューの場合

- デザインビューで **C1LinearGauge** コントロールに大目盛りを追加するには、以下の手順を実行します。
 - コントロールを右クリックし、**[プロパティ]**を選択します。
 - プロパティウィンドウで、Behavior プロパティを展開します。
 - TickMajor** プロパティを展開します。
 - Visible** を True に設定します。
- デザインビューで **C1LinearGauge** コントロールに小目盛りを追加するには、以下の手順を実行します。
 - コントロールを右クリックし、**[プロパティ]**を選択します。
 - プロパティウィンドウで、**Behavior** プロパティを展開します。
 - TickMinor** プロパティを展開します。
 - Visible を True に設定します。

ソースビューの場合

- C1LinearGauge** コントロールに大目盛りを追加するには、タグの間にマウスポインタを置き、次のマークアップを追加します。

```
<TickMajor Factor="2" Visible="True" Offset="0" Interval="10"></TickMajor>
```

- C1LinearGauge** コントロールに小目盛りを追加するには、タグの間にマウスポインタを置き、次のマークアップを追加します。

```
<TickMinor Visible="True" Offset="0" Interval="5"></TickMinor>
```

コードの場合

- C1LinearGauge** コントロールに大目盛りを追加するには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

Visual Basic

```
'C1LinearGauge に大目盛りを追加します  
C1LinearGauge1.TickMajor.Visible = True
```

C# でコードを書く場合

C#

LinearGauge for ASP.NET Web Forms

```
//C1LinearGauge に大目盛りを追加します
```

```
C1LinearGauge1.TickMajor.Visible = true;
```

- **C1LinearGauge** コントロールに小目盛りを追加するには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

Visual Basic

```
'C1LinearGauge に小目盛りを追加します
```

```
C1LinearGauge1.TickMinor.Visible = True
```

C# でコードを書く場合

C#

```
//C1LinearGauge に小目盛りを追加します
```

```
C1LinearGauge1.TickMinor.Visible = true;
```

ここまでの成果

アプリケーションを実行して、ゲージが次の図のように目盛り付きで表示されることを確認します。



対数マーカを追加する

この手順では、**C1LinearGauge** コントロールに対数マーカを追加します。以下の手順では、ページに **C1LinearGauge** コントロールが追加されていることを前提とします。

デザインビューの場合

デザインビューで **C1LinearGauge** コントロールにラベルを追加するには、以下の手順を実行します。

1. コントロールを右クリックし、**[プロパティ]**を選択します。
2. プロパティウィンドウで、**Behavior** プロパティを展開します。
3. **Islogarithmic** を True に設定します。
4. 必要に応じて、**LogarithmicBase** を設定します。

ソースビューの場合

ソースビューで **C1LinearGauge** コントロールに対数マーカを追加するには、次のように `<cc1:C1LinearGauge>` タグを変更します。

```
<cc1:C1LinearGauge ID="C1LinearGauge1" runat="server" Islogarithmic="True" LogarithmicBase="20">
```

"Islogarithmic" を **True** に設定し、必要に応じて "LogarithmicBase" を設定します。

コードの場合

C1LinearGauge コントロールに対数マーカを追加するには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

Visual Basic

```
'対数マーカを有効にします
```

```
C1LinearGauge1.Islogarithmic = True
```



```
'C1LinearGauge の LogarithmicBase を設定します
C1LinearGauge1.LogarithmicBase = 20
```

C# でコードを書く場合

```
C#
//対数マーカーを有効にします
C1LinearGauge1.Islogarithmic = true;
//C1LinearGauge の LogarithmicBase を設定します
C1LinearGauge1.LogarithmicBase = 20;
```

ここまでの成果

アプリケーションを実行して、ゲージが次の図のように表示されることを確認します。



範囲を追加する

この手順では、**C1LinearGauge** コントロールに範囲を追加します。以下の手順では、ページに **C1LinearGauge** コントロールが追加されていることを前提とします。

ソースビューの場合

C1LinearGauge コントロールに範囲を追加するには、`<cc1:C1LinearGauge>` `</cc1:C1LinearGauge>` タグの間にマウスを置き、次のマークアップを追加します。

```
<Ranges>
<cc1:GaugeRange StartValue="20" EndValue="80" StartDistance="0.85" EndDistance="0.85"
StartWidth="0.25" EndWidth="0.5">
</cc1:GaugeRange>
</Ranges>
```

範囲の位置を設定するには、`StartValue`、`EndValue`、`StartDistance`、`EndDistance`、`StartWidth`、および `EndWidth` を任意に設定します。

コードの場合

C1LinearGauge コントロールに範囲を追加するには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

```
Visual Basic
' 新しい Range オブジェクトを作成します
Dim Range1 As New C1.Web.Wijmo.Controls.C1Gauge.GaugeRange()
' 開始距離と終了距離を設定します
Range1.StartDistance = 0.85
Range1.EndDistance = 0.85
' 開始値と終了値を設定します
Range1.StartValue = 20
Range1.EndValue = 80
' 開始幅と終了幅を設定します
Range1.StartWidth = 0.25
Range1.EndWidth = 0.5
' C1LinearGauge コントロールに範囲を追加します
```

LinearGauge for ASP.NET Web Forms

```
C1LinearGauge1.Ranges.Add(Range1)
```

C# でコードを書く場合

```
C#  
  
// 新しい Range オブジェクトを作成します  
C1.Web.Wijmo.Controls.C1Gauge.GaugelRange Range1 = new C1.Web.Wijmo.Controls.C1Gauge.GaugelRange();  
// 開始距離と終了距離を設定します  
Range1.StartDistance = 0.85;  
Range1.EndDistance = 0.85;  
// 開始値と終了値を設定します  
Range1.StartValue = 20;  
Range1.EndValue = 80;  
// 開始幅と終了幅を設定します  
Range1.StartWidth = 0.25;  
Range1.EndWidth = 0.5;  
// C1LinearGauge コントロールに範囲を追加します  
C1LinearGauge1.Ranges.Add(Range1);
```

ここまでの成果

アプリケーションを実行して、ゲージが次の図のように範囲付きで表示されることを確認します。



コントロールのサイズ変更

C1LinearGauge の高さとは幅は、**Height** プロパティと **Width** プロパティを設定して簡単に変更できます。デフォルトでは、コントロールの高さは **80px** に設定され、コントロールの幅は **600px** に設定されます。コントロールの高さと幅は、ソースビュー、プロパティウィンドウ、またはコードを使用して容易に変更できます。

ソースビューの場合

ソースビューで、`Height="100px"` と `Width="500px"` を、`<cc1:C1LinearGauge>` タグ内でコントロールの高さと幅を設定したいサイズに追加します。

ソースビュー

```
<cc1:C1LinearGauge ID="C1LinearGauge1" runat="server" Height="100px" Width="500px">
```

上記の設定によって、コントロールは高さ 100 ピクセル、幅 500 ピクセルにサイズ変更されます。

プロパティウィンドウの場合

プロパティウィンドウでコントロールの高さと幅を設定するには、次のように **Height** プロパティと **Width** プロパティを変更します。

1. **C1LinearGauge** 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、必要な場合は **Layout** ノードを拡張して、**Height** プロパティと **Width** プロパティを探します。
3. **Height** の横に、コントロールの高さにするサイズの値(200px など)を入力します。

4. [幅]の横に、コントロールの幅にするサイズの値(500px など)を入力します。
5. [Enter]キーを押すか、プロパティウィンドウの外でクリックして、設定した高さと幅を **C1LinearGauge** コントロールに適用します。

コードの場合

次のコードを **Page_Load** イベントに追加し、**Height** と **Width** プロパティを 高さ 100 ピクセルと幅 500 ピクセルに設定します。

Visual Basicコードの書き方

Visual Basic

```
Me.C1LinearGauge1.Height = 100  
Me.C1LinearGauge1.Width = 500
```

C#コードの書き方

C#

```
this.C1LinearGauge1.Height = 100;  
this.C1LinearGauge1.Width = 500;
```

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数値を決定します。デフォルトでは、**C1LinearGauge** コントロールは **0** に設定された **Value** から開始しますが、デザイン時、および XAML やコードでこの数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** を設定していますが、同じ手順を使用して、他のコントロールの **Value** もカスタマイズできることに注意してください。

デザイン時の場合

C1LinearGauge コントロールの**Value** をデザイン時に設定するには、以下の手順を実行します。

1. **C1LinearGauge** を1回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て、数値(20 など)を **Value** プロパティの横のテキストボックスに入力します。これによって、**Value** プロパティが選択した数値に設定されます。

XAML の場合

たとえば、**Value**プロパティを設定するには、`Value="20"` を `<cc1:C1LinearGauge>` タグに追加して、次のような記述にします。

ソースビュー

```
<cc1:C1LinearGauge 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、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.Value = 20
```

LinearGauge for ASP.NET Web Forms

C#コードの書き方

Title Text


```
c1LinearGauge1.Value = 20;
```

✔このトピックの作業結果

最初にゲージの **Pointer** が選択した **Value** に設定されます。

最小値と最大値の設定

Min と **Max** プロパティを使用して、ゲージを制限する数値範囲を設定できます。デザイン時、および XAML やコードで、**Min** と **Max** 値をカスタマイズできます。

 **注意:** **Min** と **Max** プロパティを設定する場合、**Min** を **Max** より小さくする必要があります。また、**Value** プロパティは、**Min** と **Max** 範囲内に入る数値に設定してください(以下の例では、**Value** は以下で設定された範囲内に入る 0 に設定されています)。

デザイン時の場合

デザイン時に **C1LinearGauge** コントロールの **Min** と **Max** を設定するには、以下の手順を実行します。

1. **C1LinearGauge** を1回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て、**Max** プロパティの横に、数値(**50** など)を入力します。
3. プロパティウィンドウで、**Min** プロパティの横に、数値(**-50** など)を入力します。
これによって、**Min** と **Max** 値が設定されます。

XAML の場合

XAML で **C1LinearGauge** コントロールの **Min** と **Max** を設定するには、`Maximum="50" Minimum="-50"` を `<cc1:C1LinearGauge>` タグに追加して、次のような記述にします。

XAML の場合

```
<cc1:C1LinearGauge Name="C1LinearGauge1" Maximum="50" Minimum="-50">
```

コードの場合

C1LinearGauge コントロールの **Min** と **Max** 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;
```

✔このトピックの作業結果

実行時、ゲージは選択された範囲に制限されます。


アニメーションを設定する

この手順では、**C1LinearGauge** コントロールのアニメーションを設定します。以下の手順では、ページに **C1LinearGauge** コントロールが既に追加されていることを前提とします。

デザインビューの場合

デザインビューで **C1LinearGauge** コントロールのアニメーションを設定するには、以下の手順を実行します。

1. コントロールを右クリックし、**[プロパティ]**を選択します。
2. プロパティウィンドウで、**Behavior** プロパティを展開します。
3. **Animation** プロパティを展開します。
4. 必要に応じて、**Duration** と **Easing** を設定します。


 **メモ:** **Enabled** プロパティが **True** に設定され、**C1LinearGauge** コントロールの **Value** プロパティが設定されていることを確認します。

ソースビューの場合

ソースビューで **C1LinearGauge** コントロールのアニメーションを設定するには、`<cc1:C1LinearGauge>` `</cc1:C1LinearGauge>` タグの間にマウスを置き、次のマークアップを追加します。

```
<Animation Duration="1000" Easing ="EaseOutBounce" Enabled ="true"/>
```

必要に応じて、**Duration** と **Easing** を設定して、コントロールの **Animation** を設定します。コントロールの **Animation** を有効にするには、**Enabled** を **True** に設定します。

 **メモ:** **C1LinearGauge** コントロールの **Value** プロパティが設定されていることを確認します。

コードの場合

コントロールの **Animation** を設定するには、**Page_Load** イベントに次のコードを追加します。

Visual Basic でコードを書く場合

Visual Basic

' C1LinearGauge の最小値を設定します

```
C1LinearGauge1.Min = 0
```

' C1LinearGauge の最大値を設定します

```
C1LinearGauge1.Max = 100
```

' C1LinearGauge の値を設定します

```
C1LinearGauge1.Value = 50
```

' C1LinearGauge のアニメーションの Duration を設定します

```
C1LinearGauge1.Animation.Duration = 1000
```

' C1LinearGauge のアニメーションの Easing を設定します

```
C1LinearGauge1.Animation.Easing = C1.Web.Wijmo.Controls.C1Chart.ChartEasing.EaseOutBounce
```

' C1LinearGauge でアニメーションを有効にします

```
C1LinearGauge1.Animation.Enabled = True
```

C# でコードを書く場合

LinearGauge for ASP.NET Web Forms

C#

```
// C1LinearGauge の最小値を設定します
C1LinearGauge1.Min = 0;

// C1LinearGauge の最大値を設定します
C1LinearGauge1.Max = 100;

// C1LinearGauge の値を設定します
C1LinearGauge1.Value = 50;

// C1LinearGauge のアニメーションの Duration を設定します
C1LinearGauge1.Animation.Duration = 1000;

// C1LinearGauge のアニメーションの Easing を設定します
C1LinearGauge1.Animation.Easing = C1.Web.Wijmo.Controls.C1Chart.ChartEasing.EaseOutBounce;

// C1LinearGauge でアニメーションを有効にします
C1LinearGauge1.Animation.Enabled = true;
```