

CandlestickChart 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	3
ComponentOne for ASP.NET Web Forms のヘルプ	3
主な特長	4-5
クイックスタート	6
手順1:アプリケーションの作成	6
手順2:ローソク足チャートへの SeriesList の追加	6-8
手順3:CandlestickChart のカスタマイズ	8-9
手順4:プロジェクトの実行	9-10
ローソク足チャートのタイプ	11-13
高値 - 安値チャートを作成する	13-14
高値 - 安値 - 始値 - 終値チャートを作成する	14-15
ローソク足チャートを作成する	15-16
デザイン時のサポート	17
C1CandlestickChart スマートタグ	17-18
C1CandlestickChart のコレクションエディタ	18
CandlestickChartSeries コレクションエディタ	18-19
CandlestickChartStyle コレクションエディタ	19-20
ローソク足チャートの要素	21
軸	21-22
軸位置	22-23
軸の外観	23
正の軸値と負の軸値を表示する	23-24
軸テキストの要素	24
軸ラベルを回転する	24-25
軸ラベルを書式設定する	25-26
軸目盛りマーク	26-27
軸グリッド線	27-29
軸の範囲	29
軸の注釈	29-30
数値の注釈	30
ヘッダーとフッター	30-31

凡例	31
系列	31-32
注釈	32-33
近似曲線	34-38
エクスポートサービス	39-42

製品の概要

Candlestick Chart for ASP.NET Web Forms を使用して、ASP.NET Web アプリケーションに株価チャートを追加します。1つのコントロールで3種類のチャートを使用し、複雑な金融データも単純な金融データも豊富なカスタマイズオプションやアニメーションオプション付きで表示します。

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

CandlestickChart for ASP.NET Web Forms は、ほかにはない次の主要機能を備えています。

- HTML5 グラフ

これらのグラフが優れている理由は、SVG (Scalable Vector Graphics) を搭載していることにあります。表現力豊かでインタラクティブなビジュアル化を HTML5 で実現するには、SVG が最適です。SVG でグラフを作成する利点を挙げると、真のクライアント側でのグラフ化、DOM に適合するグラフ、ハードウェアによるグラフィックの高速化、対話形式のアニメーションとツールチップ、ストリーミングによるビジュアル化などがあります。さらに、これらのグラフにはプラグインの追加が不要です。

- データ連結

C1CandlestickChart は、クライアント側スクリプトを使用したりリモートデータソースとのデータ連結をサポートします。

- ライブデータ

すべての ComponentOne グラフは、ライブデータのストリーミングに最適化されています。グラフは、データの変化をアニメーション表示したり、遷移効果を表示しながら、データセット内のデータをビジュアル化します。

- アニメーション

棒グラフに特殊効果を追加します。遷移設定を有効にすると、ページのロード時にバウンスやエラスティックなどの遷移効果がグラフに適用されます。

- ツールチップ

ツールチップやリンクをグラフ要素に関連付けると、グラフがマウスイベントに反応するようになります。それには、HTML イメージマップを使用します。これは、クライアント側で処理されます。

- 外観

C1CandlestickChart が備える豊富なプロパティを使用して、グラフの外観(ルックアンドフィール)を変更することができます。系列ごとに異なる色を付けたり、系列内の項目ごとに異なる色を付けたり、グラフの要素に影を付けたりして、グラフの外観を変更できます。よりよいユーザーエクスペリエンスを目指して、位置、サイズ、および色も調整できます。

- 自動軸

C1CandlestickChart は、グラフの軸を自動的に生成します。

- 軸の原点

C1CandlestickChart コントロールは負の値をサポートします。各軸の **Origin** プロパティを使用して、負の値を適切に表示できます。**Origin** プロパティは、軸の開始点の描画位置を指示します。これは、正の値と負の値を同じグラフに表示する場合に適しています。

- ラベル

棒グラフにラベルを付けて読みやすくします。重要なデータポイントを強調表示したり、データやグラフに情報を提供します。

- ブラウザ間の完全な互換性

作成された UI は、すべてのブラウザとすべてのデバイスで動作します。互換性の問題を心配する必要はありません。IE6 でも iPad でもまったく同様にレンダリングされる対話式チャートを作成できます。ComponentOne ASP.NET Charts は、IE6 以上、Firefox 3 以上、Safari 3 以上、Chrome などの一般的なブラウザをサポートし

ます。

クイックスタート

CandlestickChart クイックスタートでは、**CandlestickChart** ASP.NET コントロールを初めて使用する時の手順について説明します。このクイックスタートでは、**C1CandlestickChart** コントロールを1つ含む ASP.NET アプリケーションを作成し、**C1CandlestickChart** にデータを追加し、そのコントロールをカスタマイズします。

手順1: アプリケーションの作成

このトピックでは、**C1CandlestickChart** コントロールを追加します。

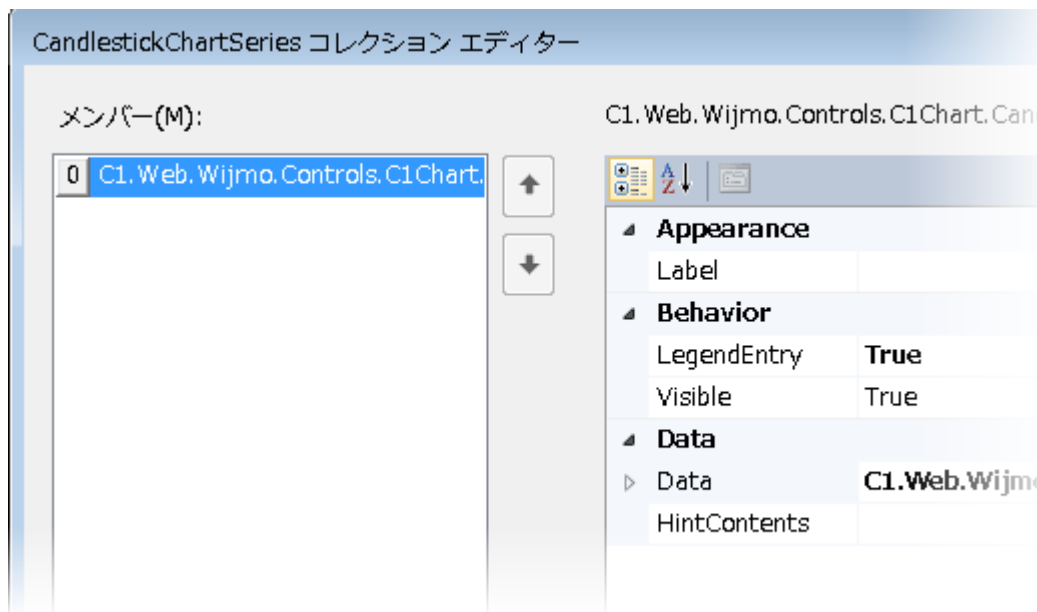
1. まず、ASP.NET Web アプリケーションを作成します。
2. プロジェクトに次の参照を追加します。
 - C1.Web.Wijmo.Controls.4.dll
 - C1.Web.Wijmo.Controls.Design.4.dll
3. ツールボックスにコントロールを追加します。
 - Visual Studio IDE (Microsoft Development Environment)を開きます。必要に応じて**[表示]**メニューの**[ツールボックス]**を選択してツールボックスを表示し、ツールボックスを右クリックしてコンテキストメニューを開きます。
 - ツールボックスに ComponentOne for ASP.NET Web Forms コンポーネントのタブを表示するには、コンテキストメニューの**[タブの追加]**を選択し、タブ名(たとえば、「ComponentOne for ASP.NET Web Forms」)を入力します。
 - コンポーネントを表示するタブを右クリックし、コンテキストメニューから**[項目の選択]**を選択します。**[ツールボックス項目の選択]**ダイアログボックスが表示されます。
 - このダイアログボックスで、**[.NET Framework コンポーネント]**タブを選択します。[名前空間]のリストをソートし(**[名前空間]**列ヘッダーをクリック)、C1.Web.Wijmo.Controls.C1Chart 名前空間に属するすべてのコンポーネントのチェックボックスをオンにします。
 - **[OK]**をクリックして、ダイアログボックスを閉じます。Visual Studio ツールボックスにコントロールが追加されます。
4. **[デザイン]**タブを選択し、Default.aspx ページからデフォルトのコンテンツを削除します。
5. デザインビューで、Visual Studio ツールボックスに移動し、**C1CandlestickChart** コントロールをダブルクリックして、ページのメインコンテンツに **C1CandlestickChart** を追加します。


手順2: ローソク足チャートへの SeriesList の追加

この手順では、**X CandlestickChartSeries** に日時値を設定し、もう一方の **CandlestickChartSeries** に倍精度値を設定します。系列のシンボルとラベルも変更します。

1. **CandlestickChart** スマートタグをクリックし、**[C1CandlestickChart のタスク]**メニューから **SeriesList** を選択します。**CandlestickChart コレクションエディタ**が開きます。

CandlestickChart for ASP.NET Web Forms



2. **[追加]**をクリックして、チャートにデータ系列を追加します。右側ペインで、**Label** プロパティを探し、データ系列に名前を付けます。
3. 次に、**Data** プロパティを探します。プロパティの横にあるドロップダウン矢印をクリックして、Data オプションを開きます。5つのオプション (Close、High、Low、Open、X) があります。
4. Close オプションの横にあるドロップダウン矢印をクリックします。DoubleValues データオプションが開きます。次の値を入力します。
 - 60, 90, 45, 50, 22, 19, 40, 33
5. High オプションの横にあるドロップダウン矢印をクリックします。DoubleValues データオプションが開きます。次の値を入力します。
 - 105, 98, 45, 50, 85, 40, 66, 92
6. Low オプションの横にあるドロップダウン矢印をクリックします。DoubleValues データオプションが開きます。次の値を入力します。
 - 8, 5, 7, 16, 20, 19, 3, 9
7. Open オプションの横にあるドロップダウン矢印をクリックします。次の値を入力します。
 - 100, 90, 32, 16, 20, 8, 17, 40
8. X オプションの横にあるドロップダウン矢印をクリックします。Value オプションの省略符ボタン  をクリックして、**ChartXData コレクションエディタ**を開きます。
9. コレクションエディタで、**[追加]**ボタンを8回クリックして X 値を追加します。左側ペインのリストから最初の項目を選択し、右側ペインでプロパティを確認します。
10. **DateTime** プロパティを見つけ、これを **03/15/2014** に設定します。
11. この手順をあと7回繰り返して、3月15日から22日までの日付を設定します。**[OK]**をクリックします。
12. **CandlestickChart コレクションエディタ**の**[OK]**をクリックすると、SeriesList データがチャートに追加されます。アプリケーションに追加されたマークアップは次のようになります。

ソースビュー

```
<cc1:C1CandlestickChart ID="C1CandlestickChart1" runat="server">
  <SeriesList>
    <cc1:CandlestickChartSeries Label="8日" LegendEntry="True">
      <Data>
        <High DoubleValues="105, 98, 45, 50, 85, 40, 66, 92"></High>
      </Data>
    </cc1:CandlestickChartSeries>
  </SeriesList>
</cc1:C1CandlestickChart>
```



```

<Low DoubleValues="8, 5, 7, 16, 20, 19, 3, 9"></Low>

<Open DoubleValues="100, 90, 32, 16, 20, 8, 17, 40"></Open>

<Close DoubleValues="60, 90, 45, 50, 22, 19, 40, 33"></Close>

<X>
  <Values>
    <cc1:ChartXData DateTimeValue="2014-03-15"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-16"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-17"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-18"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-19"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-20"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-21"></cc1:ChartXData>
    <cc1:ChartXData DateTimeValue="2014-03-22"></cc1:ChartXData>
  </Values>
</X>
</Data>
</cc1:CandlestickChartSeries>
</SeriesList>
</cc1:C1CandlestickChart>

```


この手順では、C1CandlestickChart コントロールに SeriesList データを追加しました。次の手順では、チャートの外観をカスタマイズします。

手順3: CandlestickChart のカスタマイズ

このトピックでは、軸注釈をカスタマイズします。

1. **C1CandlestickChart** スマートタグを再度クリックし、**[CandlestickChart のタスク]**メニューから**[系列のスタイル]**オプションを選択します。**CandlestickChartStyle コレクションエディタ**が開きます。
2. **[追加]**をクリックし、コレクションエディタの右側ペインで **CandlestickChartStyle** 系列を作成します。左側ペインには6つのスタイルオプションがあります。

Style	
Close	C1.Web.Wijmo.Controls.C1Chart.ChartStyle
FallingClose	C1.Web.Wijmo.Controls.C1Chart.ChartStyle
HighLow	C1.Web.Wijmo.Controls.C1Chart.ChartStyle
Open	C1.Web.Wijmo.Controls.C1Chart.ChartStyle
RaisingClose	C1.Web.Wijmo.Controls.C1Chart.ChartStyle
UnchangeClose	C1.Web.Wijmo.Controls.C1Chart.ChartStyle

3. **Close** プロパティを選択し、ドロップダウン矢印をクリックして、オプションを開きます。多数のオプションがあり、これらを使用して、**C1CandlestickChart** の外観をカスタマイズできます。このクイックスタートでは、**Fill** プロパティに注目します。**Fill** プロパティの横にあるドロップダウン矢印をクリックします。
4. **Color** プロパティをリストから探します。色の名前を入力するか、省略符ボタン  をクリックして**[その他の色]**ダイアログボックスを開き、色を選択します。
5. **Width** プロパティを見つけ、幅の数値を入力します。
6. スタイルオプションの残りのプロパティに対して上記の手順を繰り返し、**[OK]**をクリックして、スタイルをコントロールに追加します。追加されるマークアップは次のようになります。

ソースビュー

```
<CandlestickChartSeriesStyles>
<ccl:CandlestickChartStyle>
  <HighLow Width="6">
    <Fill Color="#339933"></Fill>
  </HighLow>
  <Open Width="10">
    <Fill Color="#99CCFF"></Fill>
  </Open>
  <Close Width="5">
    <Fill Color="Blue"></Fill>
  </Close>
  <FallingClose Width="9">
    <Fill Color="Fuchsia"></Fill>
  </FallingClose>
  <RaisingClose Width="8">
    <Fill Color="#660066"></Fill>
  </RaisingClose>
  <UnchangeClose Width="4">
    <Fill Color="#336600"></Fill>
  </UnchangeClose>
</ccl:CandlestickChartStyle>
</CandlestickChartSeriesStyles>
```

7. ソースビューでない場合は、ソースビューに切り替えます。次のマークアップを閉じタグ `</cc1: C1CandlestickChart>` の直前に追加します。これで、グラフの軸の外観が変更されます。

ソースビュー

```
<Axis>
  <X>
    <GridMajor Visible="True"></GridMajor>

    <GridMinor Visible="True"></GridMinor>
    <TickMajor Position="Cross">
      </TickMajor>
    </X>

    <Y Visible="False" Compass="West">
      <Labels TextAlign="Center"></Labels>

      <GridMajor Visible="True"></GridMajor>

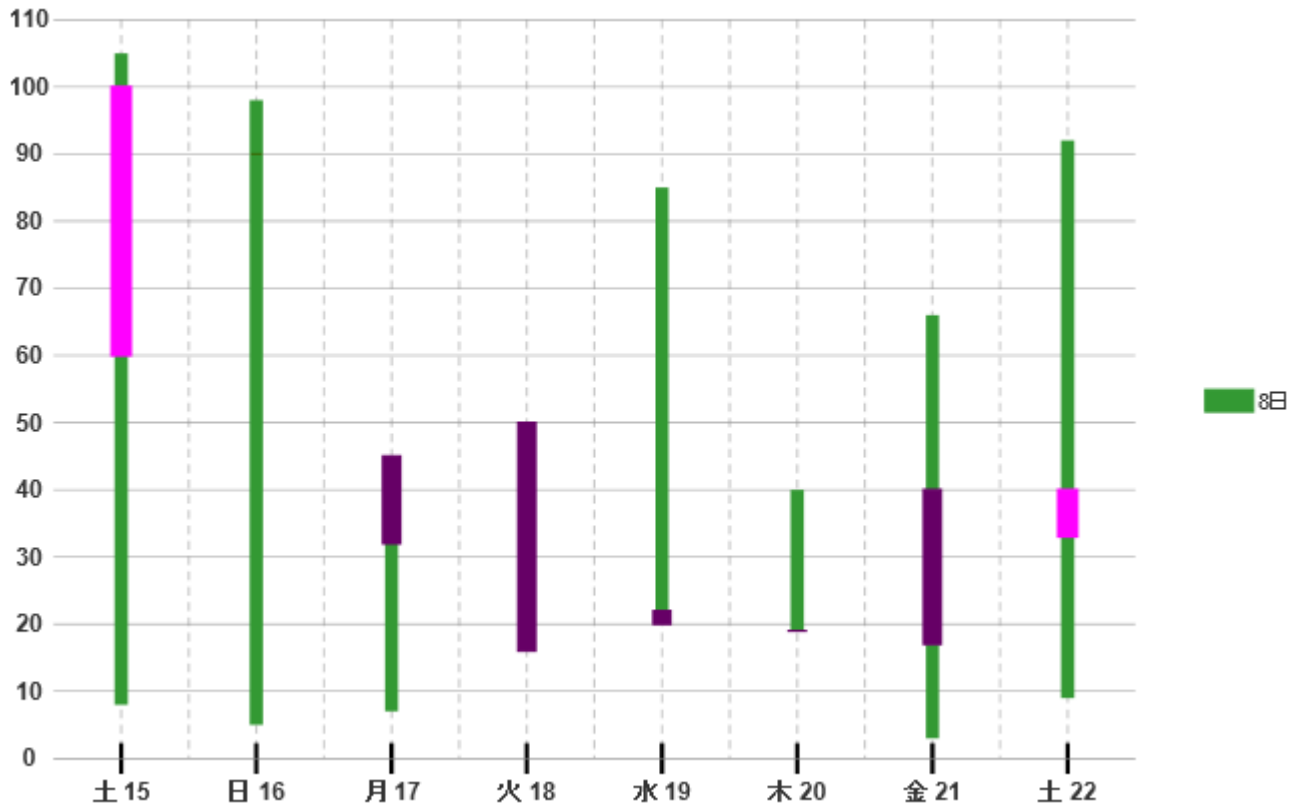
      <GridMinor Visible="False"></GridMinor>
    </Y>
  </Axis>
```

このトピックでは、**CandlestickChartStyle コレクションエディタ** とマークアップの追加を使用して、**C1CandlestickChart** コントロールをカスタマイズしました。次の手順では、プロジェクトを実行します。

手順4:プロジェクトの実行

CandlestickChart for ASP.NET Web Forms

[F5]キーを押してプロジェクトを実行します。次の図のように表示されますが、C1CandlestickChart 要素の色や幅はこの図とは異なります。

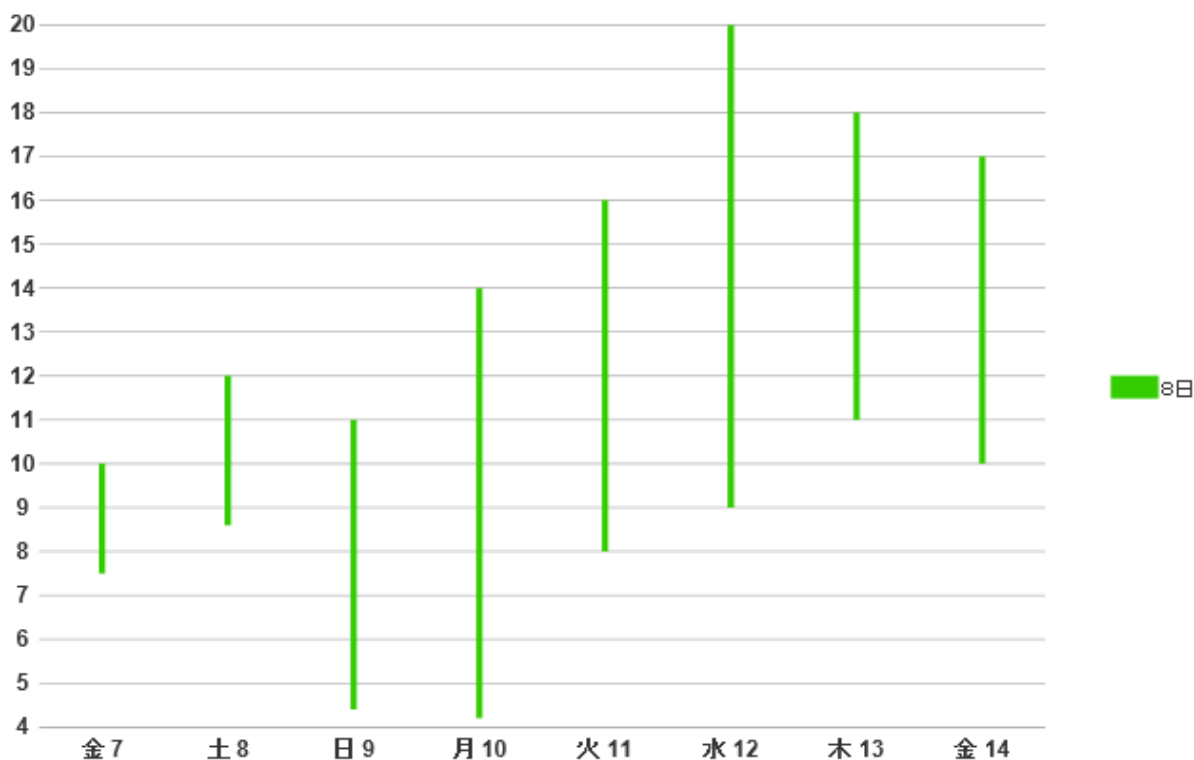


ローソク足チャートのタイプ

C1CandlestickChart コントロールでは、ローソク足チャート、高値 - 安値チャート、または高値 - 安値 - 始値 - 終値チャートの3タイプのグラフから選択できます。

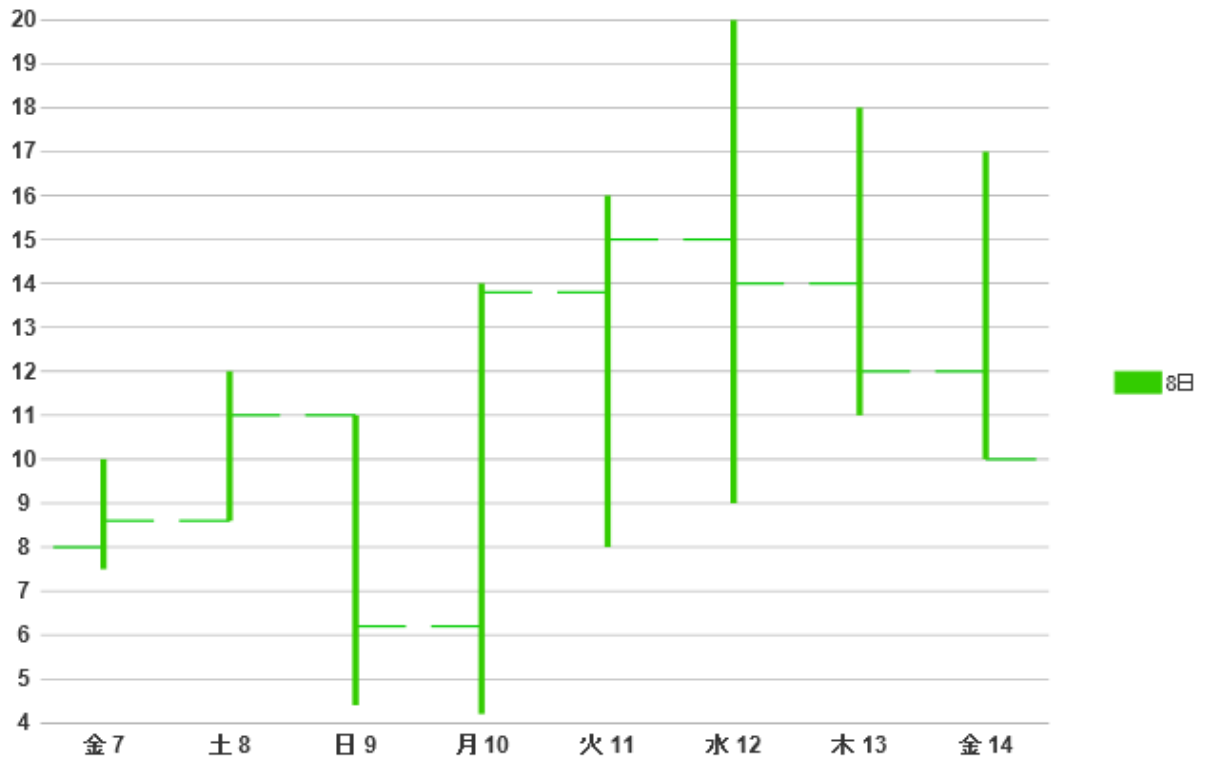
- **高値 - 安値チャート**

高値 - 安値チャートは、金融情報の表示に使用される最も簡単なグラフです。特定の項目について、1日の最高値と最安値を示します。高値 - 安値チャートは、**C1CandlestickChart** の `Type` プロパティを "HI" に設定して作成できます。



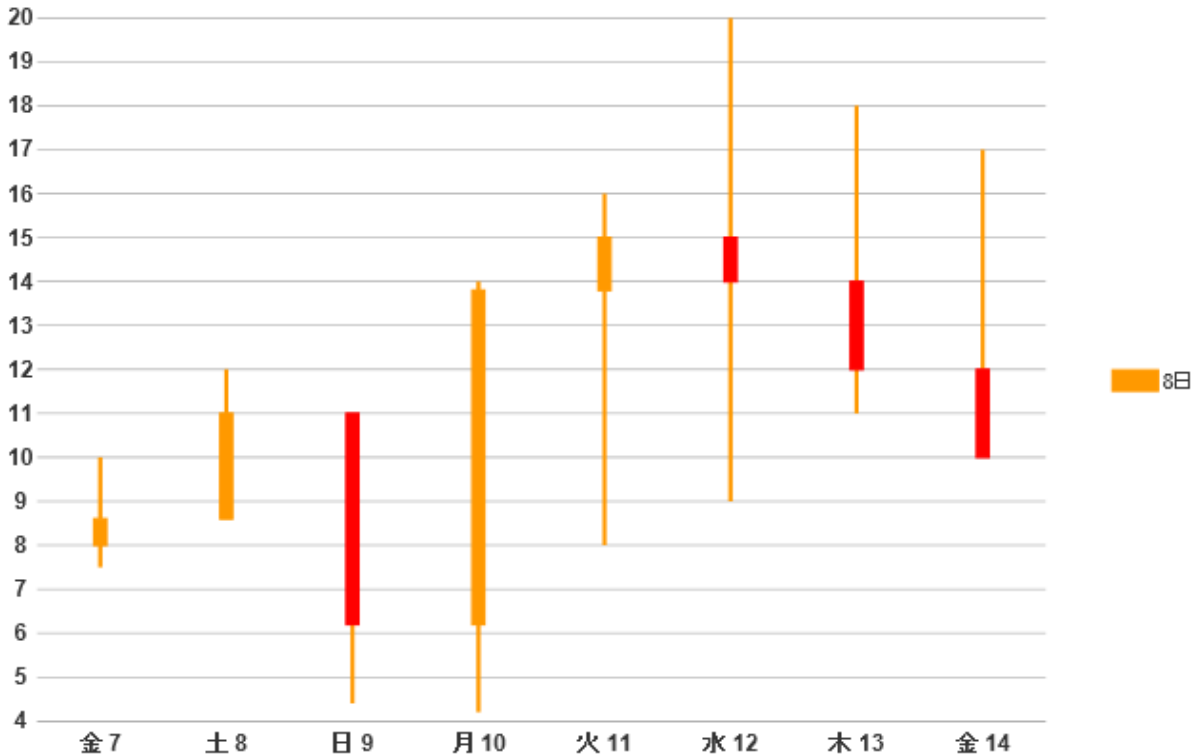
- **高値 - 安値 - 始値 - 終値チャート**

高値 - 安値 - 始値 - 終値チャートには、高値 - 安値チャートより少し多い情報が含まれます。最高値と最安値のほかに、株価の始値と終値も示します。高値 - 安値 - 始値 - 終値チャートは、**C1CandlestickChart** の `Type` プロパティを "Hloc" に設定して作成できます。



- ローソク足チャート

ローソク足チャートには、**C1CandlestickChart** コントロールを使用して作成できる金融グラフの中で最も多くの情報が含まれます。株価の高値、安値、始値、終値のほかに、株価が上昇して終了したか、下降して終了したかが示されます。このグラフでは、株価が上昇して終了したか、下降して終了したかがローソクのような棒の色で示されます。この色は、**CandlestickChartStyles** プロパティを使用してカスタマイズできます。



高値 - 安値チャートを作成する

高値 - 安値チャートは、設計時でもソースビューでも作成できます。このトピックは、適切な参照と **C1CandlestickChart** コントロールが新しいアプリケーションに追加されていることを前提とします。

設計時の場合

1. **C1CandlestickChart** コントロールを選択して、コントロールのプロパティウィンドウを開きます。
2. **Type** プロパティを **Appearance** プロパティから探します。
3. ドロップダウン矢印を使用して、"H1" をリストから選択します。
4. [クイックスタートの手順2](#)と同様に、SeriesList データを追加します。高値 - 安値チャートの場合は、高値、安値、および X 値だけが必要です。
5. **ChartStyle** オプションを使用して、チャートをカスタマイズします。
6. アプリケーションを実行します。

ソースビューの場合

次のマークアップを使用して、ソースビューで高値 - 安値チャートを作成およびカスタマイズできます。チャート要素には色と幅が設定されています。

ソースビュー

```
<cc1:C1CandlestickChart ID="C1CandlestickChart1" runat="server" Type="H1">
  <CandlestickChartSeriesStyles>
    <cc1:CandlestickChartStyle>
      <HighLow Width="3">
```

```

        <Fill Color="#0033CC">
        </Fill>
    </HighLow>
</cc1:CandlestickChartStyle>
</CandlestickChartSeriesStyles>
<SeriesList>
    <cc1:CandlestickChartSeries LegendEntry="True" Label="High-Low Values">
        <Data>
            <High DoubleValues="10, 12, 11, 14, 16, 20, 18, 17" />
            <Low DoubleValues="7.5, 8.6, 4.4, 4.2, 8, 9, 11, 10" />
            <X>
                <Values>
                    <cc1:ChartXData DateTimeValue="2014-03-07" />
                    <cc1:ChartXData DateTimeValue="2014-03-08" />
                    <cc1:ChartXData DateTimeValue="2014-03-09" />
                    <cc1:ChartXData DateTimeValue="2014-03-10" />
                    <cc1:ChartXData DateTimeValue="2014-03-11" />
                    <cc1:ChartXData DateTimeValue="2014-03-12" />
                    <cc1:ChartXData DateTimeValue="2014-03-13" />
                    <cc1:ChartXData DateTimeValue="2014-03-14" />
                </Values>
            </X>
        </Data>
    </cc1:CandlestickChartSeries>
</SeriesList>
</cc1:C1CandlestickChart>

```

高値 - 安値 - 始値 - 終値チャートを作成する

高値 - 安値 - 始値 - 終値チャートは、設計時でもソースビューでも作成できます。このトピックは、適切な参照と **C1CandlestickChart** コントロールが新しいアプリケーションに追加されていることを前提とします。

設計時の場合

1. **C1CandlestickChart** コントロールを選択して、コントロールのプロパティウィンドウを開きます。
2. **Type** プロパティを **Appearance** プロパティから探します。
3. ドロップダウン矢印を使用して、"Hloc" をリストから選択します。
4. [クイックスタートの手順2](#)と同様に、SeriesList データを追加します。高値 - 安値 - 始値 - 終値チャートの場合は、高値、安値、始値、終値、および X 値が必要です。
5. **ChartStyle** オプションを使用して、チャートをカスタマイズします。
6. アプリケーションを実行します。

ソースビューの場合

次のマークアップを使用して、ソースビューで高値 - 安値 - 始値 - 終値チャートを作成およびカスタマイズできます。チャート要素には色と幅が設定されています。

ソースビュー

```

<cc1:C1CandlestickChart ID="C1CandlestickChart1" runat="server" Type="Hloc">
    <CandlestickChartSeriesStyles>
        <cc1:CandlestickChartStyle>
            <HighLow Width="3">

```

CandlestickChart for ASP.NET Web Forms

```
<Fill Color="#0033CC">
</Fill>
</HighLow>
</cc1:CandlestickChartStyle>
</CandlestickChartSeriesStyles>
<SeriesList>
  <cc1:CandlestickChartSeries LegendEntry="True" Label="Values">
    <Data>
      <High DoubleValues="10, 12, 11, 14, 16, 20, 18, 17" />
      <Low DoubleValues="7.5, 8.6, 4.4, 4.2, 8, 9, 11, 10" />

      <Open DoubleValues="8, 8.6, 11, 6.2, 13.8, 15, 14, 12"></Open>

      <Close DoubleValues="8.6, 11, 6.2, 13.8, 15, 14, 12, 10"></Close>
      <X>
        <Values>
          <cc1:ChartXData DateTimeValue="2014-03-07" />
          <cc1:ChartXData DateTimeValue="2014-03-08" />
          <cc1:ChartXData DateTimeValue="2014-03-09" />
          <cc1:ChartXData DateTimeValue="2014-03-10" />
          <cc1:ChartXData DateTimeValue="2014-03-11" />
          <cc1:ChartXData DateTimeValue="2014-03-12" />
          <cc1:ChartXData DateTimeValue="2014-03-13" />
          <cc1:ChartXData DateTimeValue="2014-03-14" />
        </Values>
      </X>
    </Data>
  </cc1:CandlestickChartSeries>
</SeriesList>
</cc1:C1CandlestickChart>
```

ローソク足チャートを作成する

ローソク足チャートは、設計時でもソースビューでも作成できます。このトピックは、適切な参照と **C1CandlestickChart** コントロールが新しいアプリケーションに追加されていることを前提とします。"Candlestick" がデフォルトのグラフタイプです。

設計時の場合

1. **C1CandlestickChart** コントロールを選択して、コントロールのプロパティウィンドウを開きます。
2. **Type** プロパティを **Appearance** プロパティから探します。
3. タイプをチェックして、"Candlestick" が選択されていることを確認します。これはデフォルトのグラフタイプなので、自動的に選択されます。
4. [クイックスタートの手順2](#)と同様に、SeriesList データを追加します。ローソク足チャートの場合は、高値、安値、始値、終値、および X 値が必要です。
5. **ChartStyle** オプションを使用して、チャートをカスタマイズします。
6. アプリケーションを実行します。

ソースビューの場合

次のマークアップを使用して、ソースビューでローソク足チャートを作成およびカスタマイズできます。チャート要素には色と幅が設定されています。

ソースビュー

```

<cc1:C1CandlestickChart ID="C1CandlestickChart1" runat="server" Type="Candlestick">
  <CandlestickChartSeriesStyles>
    <cc1:CandlestickChartStyle>
      <HighLow Fill-Color="#ff9900" Width="2"></HighLow>
      <FallingClose Fill-Color="#ff0000" Width="6"></FallingClose>
      <RaisingClose Fill-Color="#ff9900" Width="6"></RaisingClose>
    </cc1:CandlestickChartStyle>
  </CandlestickChartSeriesStyles>
  <SeriesList>
    <cc1:CandlestickChartSeries LegendEntry="True" Label="Values">
      <Data>
        <High DoubleValues="10, 12, 11, 14, 16, 20, 18, 17" />
        <Low DoubleValues="7.5, 8.6, 4.4, 4.2, 8, 9, 11, 10" />
        <Open DoubleValues="8,8.6,11,6.2,13.8,15,14,12"></Open>
        <Close DoubleValues="8.6,11,6.2,13.8,15,14,12,10"></Close>
        <X>
          <Values>
            <cc1:ChartXData DateTimeValue="2014-03-07" />
            <cc1:ChartXData DateTimeValue="2014-03-08" />
            <cc1:ChartXData DateTimeValue="2014-03-09" />
            <cc1:ChartXData DateTimeValue="2014-03-10" />
            <cc1:ChartXData DateTimeValue="2014-03-11" />
            <cc1:ChartXData DateTimeValue="2014-03-12" />
            <cc1:ChartXData DateTimeValue="2014-03-13" />
            <cc1:ChartXData DateTimeValue="2014-03-14" />
          </Values>
        </X>
      </Data>
    </cc1:CandlestickChartSeries>
  </SeriesList>
</cc1:C1CandlestickChart>

```

デザイン時のサポート


C1CandlestickChart は、カスタマイズされたコンテキストメニュー、スマートタグ、およびデザイナを備え、豊富な設計時サポートを提供し、オブジェクトモデルの操作を容易にします。

以下のセクションでは、**C1CandlestickChart** の設計時環境を使用して、**C1CandlestickChart** コントロールを設定する方法について説明します。

C1CandlestickChart スマートタグ

Visual Studio では、**C1CandlestickChart** コントロールにスマートタグが含まれます。スマートタグは、**C1BarChart** で最もよく使用されるプロパティを提供するショートカットタスクメニューです。

C1CandlestickChart コントロールは、スマートタグを使用して、よく使用するプロパティにすばやく簡単にアクセスできます。

[**C1CandlestickChart のタスク**]メニューにアクセスするには、**C1CandlestickChart** コントロールの右上隅にあるスマートタグ  をクリックします。[**C1CandlestickChart のタスク**]メニューが表示されます。



[**C1CandlestickChart のタスク**]メニューの機能は次のとおりです。

- **データソースの選択**

[**データソースの選択**]項目をクリックすると、ドロップダウンリストが開きます。ここで、連結先として、既存のデータソースを選択するか、新しいデータソースを選択します。

- **SeriesList**

SeriesList 項目を選択すると、[**CandlestickChartSeries コレクションエディタ**]ダイアログボックスが開きます。ここで、**C1CandlestickChart** コントロールの **CandlestickChartSeries** メンバを追加または削除します。**CandlestickChartSeries** メンバを追加したら、メンバのプロパティを変更できます。

- **SeriesStyles**

SeriesStyles 項目をクリックすると、[**CandlestickChartStyle コレクションエディタ**]ダイアログボックスが開きます。ここで、**C1CandlestickChart** コントロールの **ChartStyle** メンバを追加または削除します。**ChartStyle** メンバを追加したら、メンバのプロパティを変更します。

- **SeriesHoverStyles**

SeriesHoverStyle 項目をクリックすると、[**CandlestickChartStyle コレクションエディタ**]ダイアログボックスが開きます。ここで、**C1CandlestickChart** コントロールの **ChartStyle** メンバを追加または削除します。**ChartStyle** メンバを追加したら、メンバのプロパティを変更します。

- **CDN の使用**

[**CDN の使用**]チェックボックスをオンにすると、クライアントリソースが CDN からロードされます。デフォルトでは、このチェックボックスはオフです。

- **CDN パス**

CDN の URL パスを表示します。

- **バージョン情報**

[**バージョン情報**]項目をクリックすると、[**ComponentOne for ASP.NET Web Forms のバージョン情報**]ダイアログボックスが表示され、**ComponentOne for ASP.NET Web Forms** のバージョン番号およびオンラインリソースが表示されます。

C1CandlestickChart のコレクションエディタ

C1CandlestickChart には、SeriesList データやスタイルを追加したり削除するために、次のコレクションエディタが付属しています。

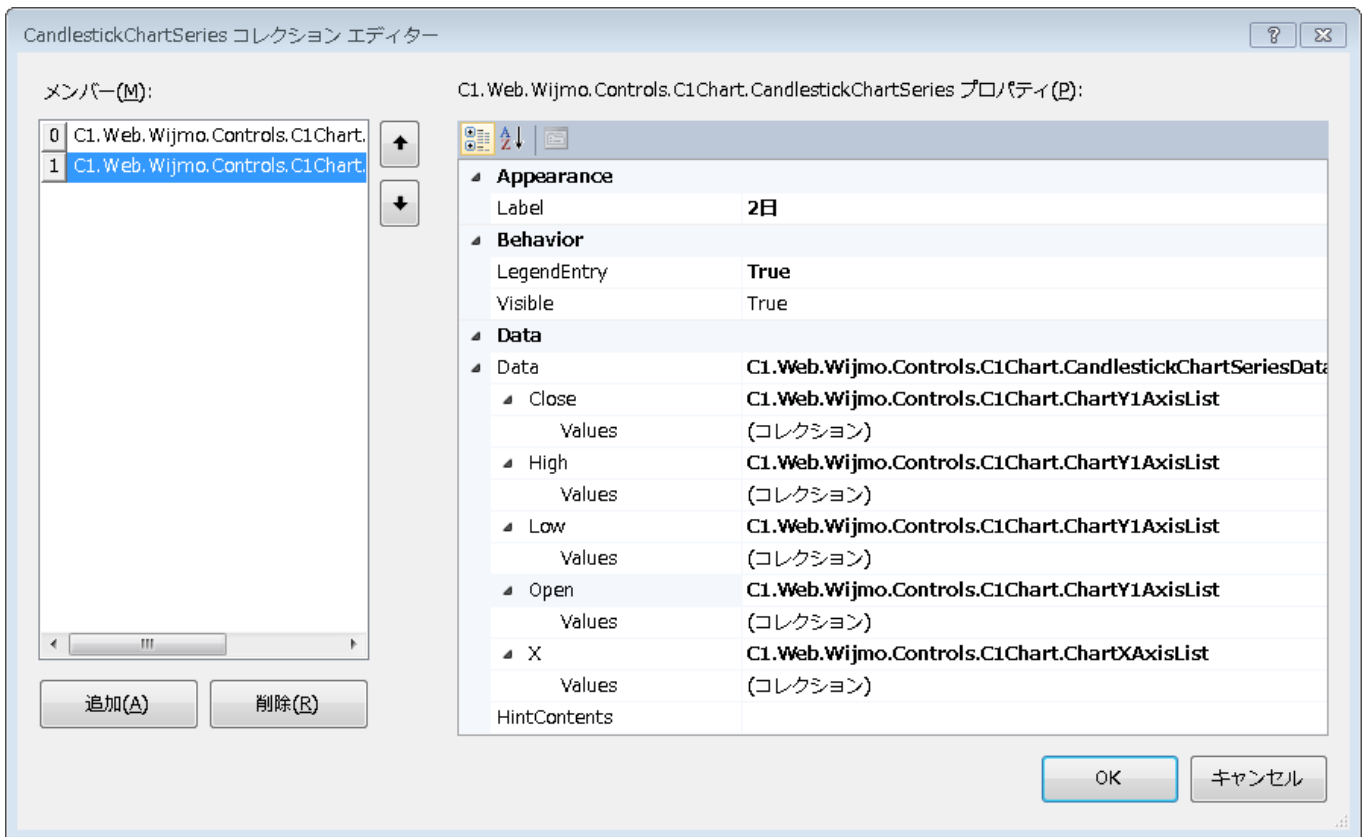
- **CandlestickChartSeries コレクションエディタ**
- **CandlestickChartStyle コレクションエディタ**

CandlestickChartSeries コレクションエディタ

CandlestickChartSeries コレクションエディタを使用して、選択した **CandlestickChartSeries** メンバを **C1CandlestickChart** コントロールに追加したり、コントロールから削除することができます。**CandlestickChartSeries** メンバを追加したら、メンバのプロパティを変更できます。

[**追加**]ボタンをクリックして新しいメンバを系列に追加すると、**CandlestickChartSeries コレクションエディタ**は、次のように表示されます。

CandlestickChart for ASP.NET Web Forms

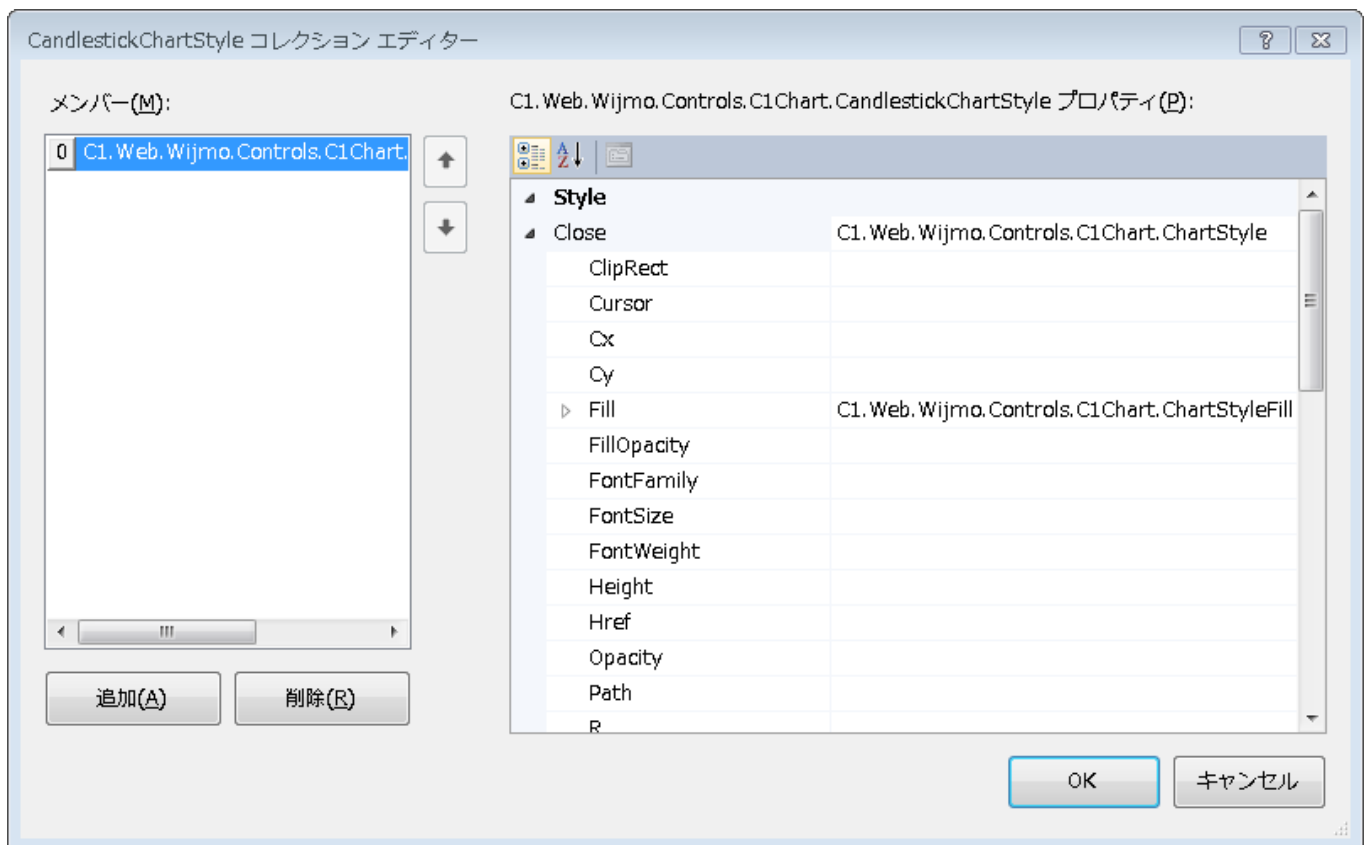


CandlestickChartSeries コレクションエディタにアクセスするには

1. **C1CandlestickChart** コントロールを選択し、スマートタグをクリックします。
2. **C1CandlestickChart** タスクメニューで、**SeriesList** 項目を選択します。[CandlestickChartSeries コレクションエディタ]ダイアログボックスが開きます。

CandlestickChartStyle コレクションエディタ

CandlestickChartStyle コレクションエディタを使用して、C1CandlestickChart コントロールの ChartStyle メンバを追加または削除できます。ChartStyle メンバを追加したら、メンバのプロパティを変更します。



CandlestickChartStyle コレクションエディタにアクセスするには

1. **C1CandlestickChart** コントロールを選択し、スマートタグをクリックします。
2. **C1CandlestickChart** タスクメニューで、**SeriesStyle** 項目を選択すると、**[CandlestickChartStyle コレクションエディタ]**ダイアログボックスが開きます。

ローソク足チャートの要素

このセクションでは、**C1CandlestickChart** コントロールを構成する要素について画像を使用してわかりやすく説明します。トピックは、**C1CandlestickChart** コントロールのさまざまな側面を表す要素ごとに分かれています。

軸

ChartAxes オブジェクトの X プロパティと Y プロパティは、チャート軸の外観をカスタマイズするための **ChartAxis** オブジェクトを返します。軸は、**ChartAxes** プロパティのサブプロパティである **ChartAxes.X** と **ChartAxes.Y** によって表されます。どちらのプロパティも、次の主要なプロパティを持つ **ChartAxis** オブジェクトを返します。

軸プロパティ	説明
ChartAxis.Alignment	軸テキストの配置を示す値。このプロパティのデフォルト値は ChartAxisAlignment.Center です。
ChartAxis.AnnoFormatString	注釈の書式文字列を示す値。
ChartAxis.AnnoMethod	注釈のメソッドを示す値。このプロパティのデフォルト値は ChartAxisAnnoMethod.Values です。
ChartAxis.AutoMajor	大目盛りマークの値を自動的に計算するかどうかを示す値。このプロパティのデフォルト値は true です。
ChartAxis.AutoMax	軸の最大値を自動的に計算するかどうかを示す値。
ChartAxis.AutoMin	軸の最小値を自動的に計算するかどうかを示す値。このプロパティのデフォルト値は True です。
ChartAxis.AutoMinor	小目盛りマークの値を自動的に計算するかどうかを示す値。このプロパティのデフォルト値は true です。
ChartAxis.Compass	軸の方位を示す値。このプロパティのデフォルト値は ChartCompass.South です。
ChartAxis.GridMajor	主グリッド線の情報を提供する値。
ChartAxis.GridMinor	副グリッド線の情報を提供する値。

ChartAxis.Labels	ラベルの情報を提供する値。
ChartAxis.Max	軸の最大値を示す値。このプロパティのデフォルト値は 0 です。
ChartAxis.Min	軸の最小値を示す値。このプロパティのデフォルト値は 0 です。
ChartAxis.Origin	軸の原点値を示す値。
ChartAxis.Style	軸のスタイルを示す値。
ChartAxis.Text	軸のテキストを示す値。
ChartAxis.TextStyle	軸のテキストのスタイルを示す値。
ChartAxis.TextVisible	軸テキストの表示/非表示を示す値。このプロパティのデフォルト値は True です。
ChartAxis.TickMajor	大目盛りの情報を提供する値。
ChartAxis.TickMinor	小目盛りの情報を提供する値。
ChartAxis.UnitMajor	大目盛りマーク間のきざみを示す値。このプロパティのデフォルト値は 0 です。
ChartAxis.UnitMinor	小目盛りマーク間のきざみを示す値。このプロパティのデフォルト値は 0 です。
ChartAxis.ValueLabels	軸の valueLabel のコレクションを示す値。
ChartAxis.Visible	軸の表示/非表示を示す値。このプロパティのデフォルト値は True です。

軸位置

CandlestickChart for ASP.NET Web Forms

プロパティを1つ使用して、グラフ軸の位置を指定できます。**ChartAxis.Compass** は、方位 (North、South、East、West) を使用して軸要素の位置を設定します。

- **X 軸**

X 軸の **ChartAxis.Compass** プロパティは、軸の位置をチャートの上部 (North) またはチャートの下部 (South) に設定します。デフォルトでは、**ChartAxis.Compass** プロパティは South に設定されます。

- **Y 軸**

Y 軸の **ChartAxis.Compass** プロパティは、軸の位置をチャートの左側 (West) または右側 (East) に設定します。デフォルトでは、**ChartAxis.Compass** プロパティは West に設定されます。

軸の外観

ChartAxis.Alignment プロパティには、3つの設定値 (**Center**、**Near**、**Far**) を設定できます。配置を Center に設定すると、軸タイトルは、チャート領域の中央に配置されます。配置を Near に設定すると、軸タイトルは、チャート領域の左側に配置されます。配置を Far に設定すると、軸タイトルは、チャート領域の右側に配置されます。

X 軸ラベルを設計時に変更するには、**Axis -> X -> Labels -> AxisLabelStyle** の順にノードを展開し、**FontSize** プロパティを設定します。

Fill.Color プロパティは、軸線、目盛りマーク、ラベル、およびタイトルの色を変更します。たとえば、X 軸ラベルのフォントの色を設計時に変更するには、**Axis -> X -> Labels -> AxisLabelStyle -> Fill** の順にノードを展開し、**[色]** の横にある省略符ボタンをクリックして、色を選択します。

次の例は、いくつかのプロパティを設定した後のソースビューを示します。

ソースビュー

```
<Axis>

    <X Max="2010" AutoMin ="false" Min="2005" Text="Year">

        <TextStyle FontSize="16">

            </TextStyle>

            <TickMajor Position="Outside">

                </TickMajor>

            </X>
        </Axis>
```

正の軸値と負の軸値を表示する

Origin プロパティを使用して、軸の開始点の描画位置を指示します。これは、正の値と負の値を同じグラフに表示する場合に適しています。

- **Origin プロパティをデザインビューで設定するには**

1. プロジェクトに **C1.Web.Wijmo.Controls.Design.4.dll** への参照を追加します。
2. **C1CandlestickChart** をツールボックスからページに追加します。C1CandlestickChart アイコンをツールボックス

に追加する方法の詳細については、「[ASP.NET Web Forms の開始](#)」を参照してください。

3. C1CandlestickChart プロパティウィンドウで、**Axis -> X** の順にノードを展開し、Origin プロパティを 2.6 に設定します。
4. C1CandlestickChart プロパティウィンドウで、**Axis -> Y** の順にノードを展開し、**Origin** プロパティを **0** に、**Alignment** を **Far** にそれぞれ設定します。

- **Origin プロパティをソースビューで設定するには**

1. プロジェクトに **C1.Web.Wijmo.Controls.Design.4.dll** への参照を追加します。
2. **C1CandlestickChart** をツールボックスからページに追加します。C1CandlestickChart アイコンをツールボックスに追加する方法の詳細については、「[ASP.NET Web Forms の開始](#)」を参照してください。
3. [ソース]タブをクリックし、次のコードを <Axis> タグ内に追加します。

ソースビュー

```
<Axis>
  <X Origin="2.6" ></X>
  <Y Alignment="Far" Visible="true" Origin="0" ></Y>
</Axis>
```

軸テキストの要素

軸にタイトルを付加すると、その軸に沿って何がチャート化されているかが明確になります。軸のタイトルや注釈を回転することもできます。

- **軸タイトルを付加するには**

軸にタイトルを付加するには、軸の **Text** プロパティを使用します。タイトルを削除するには、テキストを Text プロパティから削除します。

- **軸のタイトルと注釈を回転するには**

軸タイトルを 90、180、または 270 度に回転するには、**Rotation** プロパティを使用します。垂直軸には、90 度と 270 度の回転が最も効果的です。X 軸ラベルの回転を設計時に変更するには、**Axis -> X -> Labels -> AxisLabelStyle** の順にノードを展開し、**Rotation** プロパティを設定します。

- **テキスト要素を書式設定するには**

C1CandlestickChart のテキスト要素を書式設定するには、Text プロパティを使用します。テキスト要素を設計時に変更するには、**Axis -> X -> Labels** の順にノードを展開し、Text プロパティを設定します。

軸ラベルを回転する

このトピックでは、Rotation プロパティを使用して、設計時およびプログラムで軸ラベルを回転する方法を示します。

- **設計時に Y 軸ラベルを設定するには**

1. 設計時に、CandlestickChart コントロールを選択し、プロパティウィンドウに移動します。
2. Axis -> X -> Labels -> AxisLabelStyle の順にノードを展開し、Rotation プロパティを 60 に設定します。

- **Y 軸ラベルをプログラムで設定するには**

Visual Basic コードの書き方

CandlestickChart for ASP.NET Web Forms

Visual Basic

```
CandlestickChart.Axis.X.Labels.AxisLabelStyle.Rotation = 60
```

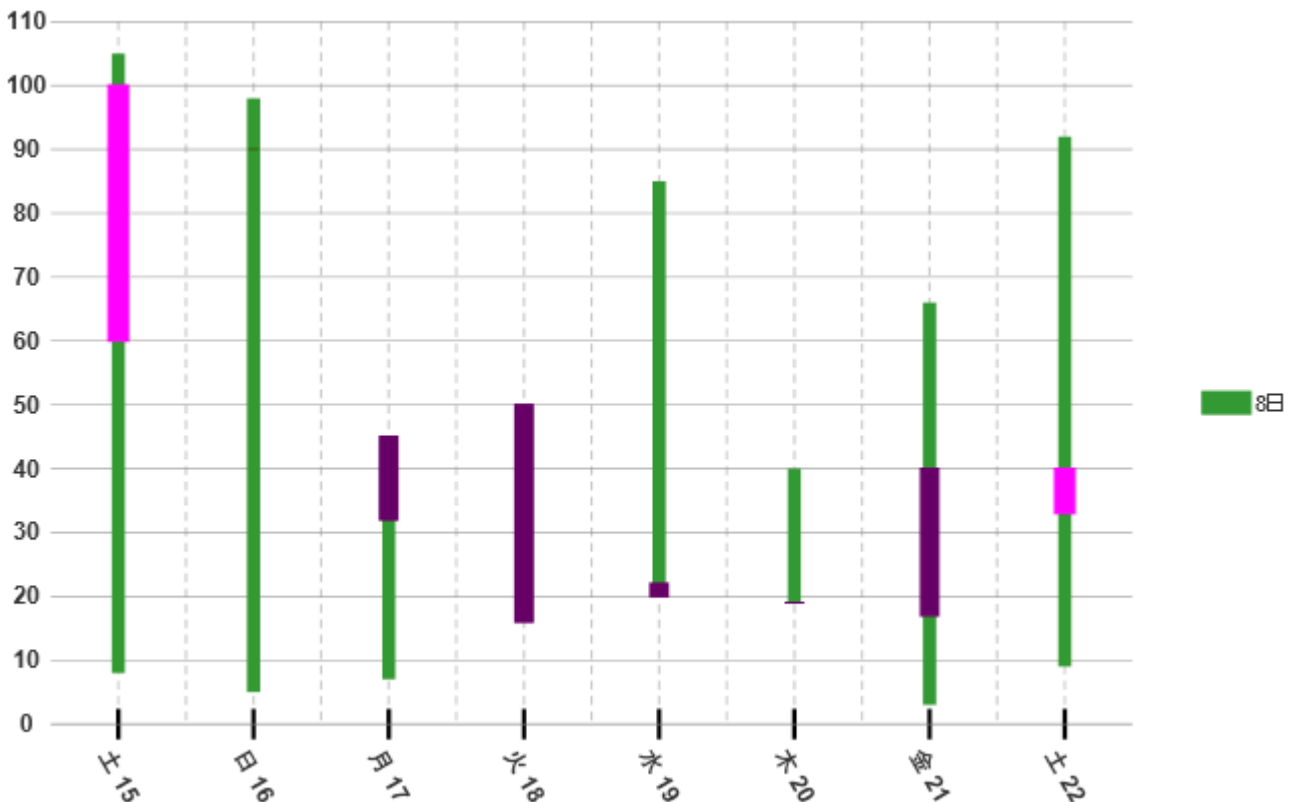
C# コードの書き方

C#

```
CandlestickChart.Axis.X.Labels.AxisLabelStyle.Rotation = 60;
```

このトピックの作業結果

次の図は、60 度回転した X 軸ラベルを示します。




軸ラベルを書式設定する

BarChart の軸ラベルは、Labels.Width、Labels.TextAlign、Labels.Style.Rotation の各プロパティを使用して書式設定できます。

● 軸ラベルの幅および配置をデザインビューで書式設定するには

1. プロジェクトに **C1.Web.Wijmo.Controls.4.dll** への参照を追加します。
2. **C1CandlestickChart** をツールボックスからページに追加します。**C1CandlestickChart** アイコンをツールボックスに追加する方法の詳細については、「[ASP.NET Web Forms の開始](#)」を参照してください。
3. **C1CandlestickChart** プロパティウィンドウで、X -> Labels の順にノードを展開し、TextAlign プロパティを Center に、Width プロパティを 150 にそれぞれ設定します。

 これにより、X 軸のラベルが中央に配置され、各ラベルが固定幅 150 で作成されます。

- グラフラベルの幅および配置をソースビューで書式設定するには

グラフラベルの幅と配置をソースビューで書式設定するには、[ソース]タブをクリックし、次のコードをソースファイルに追加します。

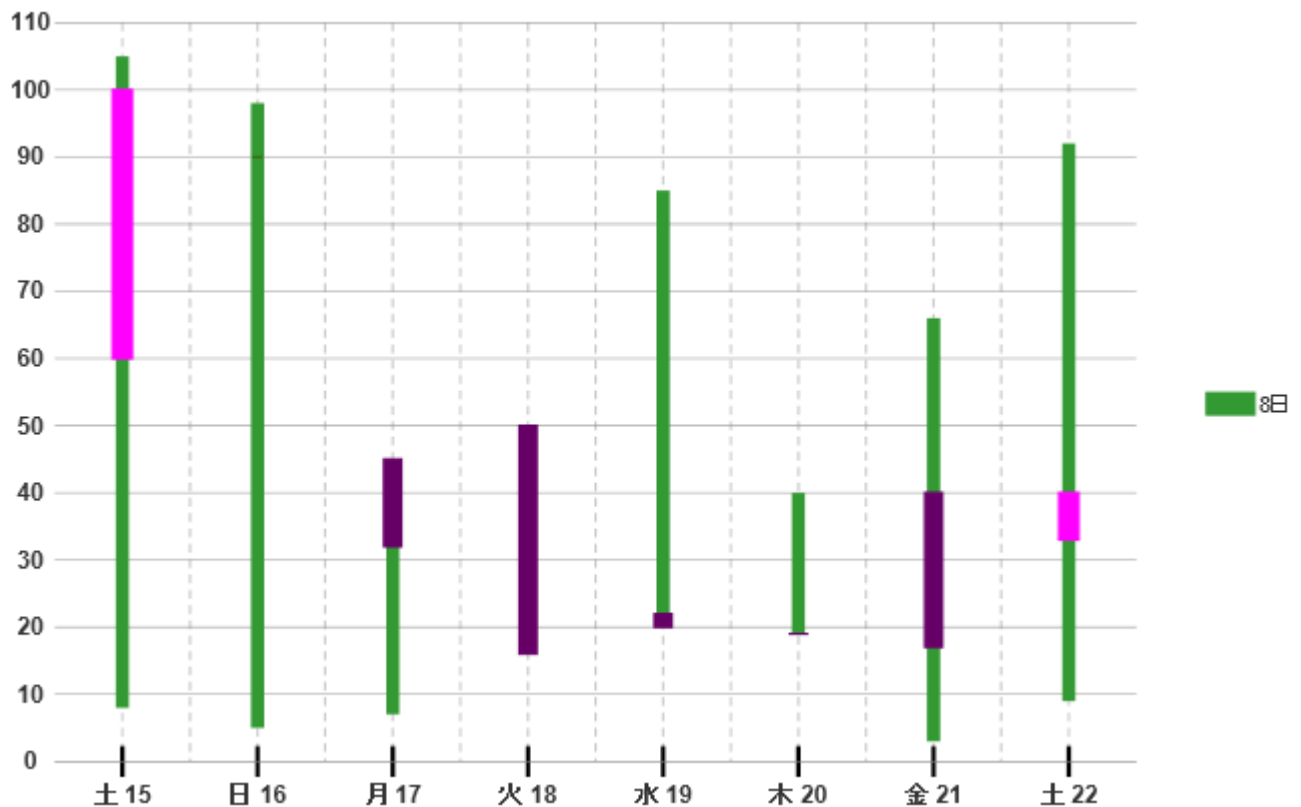
<X></X> タグ内に、次のマークアップを追加します。

マークアップ

```
<Labels Width="150" TextAlign="Center">
<Style Rotation="0"></Style>
```

このトピックの作業結果

これにより、X 軸のラベルが中央に配置され、各ラベルが固定幅 150 で作成されます。



軸目盛りマーク

目盛りマークは、測定単位をチャートに表示するために、軸に垂直に表示される線です。**ChartAxisTick.Position** を Cross または Outside に設定すると、大小の目盛りマークをバブルチャートに表示できます。目盛りのスペーシングまたは目盛りの属性のカスタマイズは、一連のプロパティを操作するだけで、極めて簡単に実行できます。

Axis.TickMajor プロパティと **Axis.TickMinor** プロパティは、軸の目盛りマークの状態を設定します。このプロパティは、**ChartAxisTickPosition** のいずれかの値に設定できます。

- 目盛りマークの位置

CandlestickChart for ASP.NET Web Forms

以下の値は、目盛りマークの位置および目盛りマークを表示するかどうかを設定します。

値	説明
ChartAxisTickPosition.None	軸に目盛りマークを表示しません。
ChartAxisTickPosition.Cross	軸に交差する目盛りマークを表示します。
ChartAxisTickPosition.Outside	軸に沿ってチャート領域の外側に目盛りマークを表示します。
ChartAxisTickPosition.Inside	軸に沿ってチャート領域の内側に目盛りマークを表示します。

● 目盛りマークのスペーシング

ChartAxis.AutoMajor プロパティと **ChartAxis.AutoMinor** プロパティは、チャートの目盛りマークを自動的に構成するかどうかを設定します。両方のプロパティを **True** に設定すると、現在のデータを使用して大小の目盛りマークが論理的に配置されます。**ChartAxis.AutoMajor** プロパティが true の場合、軸の注釈の重なりを有効にする必要はありません。

ChartAxis.UnitMajor プロパティと **ChartAxis.UnitMinor** プロパティは、目盛りのきざみ幅を設定します。**ChartAxis.UnitMajor** プロパティを設定すると、**ChartAxis.UnitMinor** プロパティの値は自動的に UnitMajor の半分の値に設定されます。**ChartAxis.UnitMinor** プロパティは自動的に設定されますが、手動で別の値に設定することもできます。

● 目盛りマークの長さ

大小の目盛りマークの長さは、**ChartAxisTick.Factor** プロパティで拡大することができます。**ChartAxisTick.Factor** プロパティを使用するには、その前に、**ChartAxisTick.Position** を **Outside** または **Cross** に設定します。目盛りマークのサイズは、この目盛り係数と軸線の太さに基づいて設定されます。目盛り係数を2倍にすると、軸の目盛りマークの長さが2倍になります。X 軸の目盛りマークに負の値を使用すると、目盛りマークは、X 軸ラベルの上に表示されます。Y 軸の目盛りマークに負の値を使用すると、目盛りマークは、Y 軸ラベルの左側に表示されます。



ChartAxisTick.Factor プロパティの値を大きくすると、**MarginBottom**、**MarginLeft**、**MarginRight**、または **MarginTop** プロパティも、拡大された目盛りマークの長さに合わせて十分な間隔に拡大されます。

軸グリッド線

グリッド線は、大/小間隔を単位として大/小目盛りマークから垂直に表示されます。軸に垂直に大間隔で表示される線は、**ChartAxis.GridMajor** プロパティによって制御され、軸に垂直に小間隔で表示される線は、**ChartAxis.GridMinor** プロパティによって制御されます。グリッド線は、正確な値が必要な場合にチャートを読みやすくすることができます。

次のマークアップに示すように、GridMajor プロパティと GridMinor プロパティを X 軸と Y 軸の両方に設定できます。

上記のマークアップで設定されたプロパティは、次の図のようなグラフになります。この図では、X 軸に主グリッド線と副グリッド線の両方が表示されます。GridMajor 線は実線、GridMinor 線は破線です。

ソースビュー

```
<Axis>
  <X>
    <GridMajor Visible="True">
      <Style Stroke="#0099FF" StrokeDashArray="2,2" StrokeLineCap="bevel"
StrokeLineJoin="round"></Style>

      <GridStyle Stroke="#0099FF" StrokeDashArray="2,2" StrokeLineCap="bevel"
StrokeLineJoin="round"></GridStyle>
    </GridMajor>
    <GridMinor Visible="True">
      <Style Stroke="#0099FF">
        </Style>
      <GridStyle Stroke="#0099FF">
        </GridStyle>
    </GridMinor>
  </X>

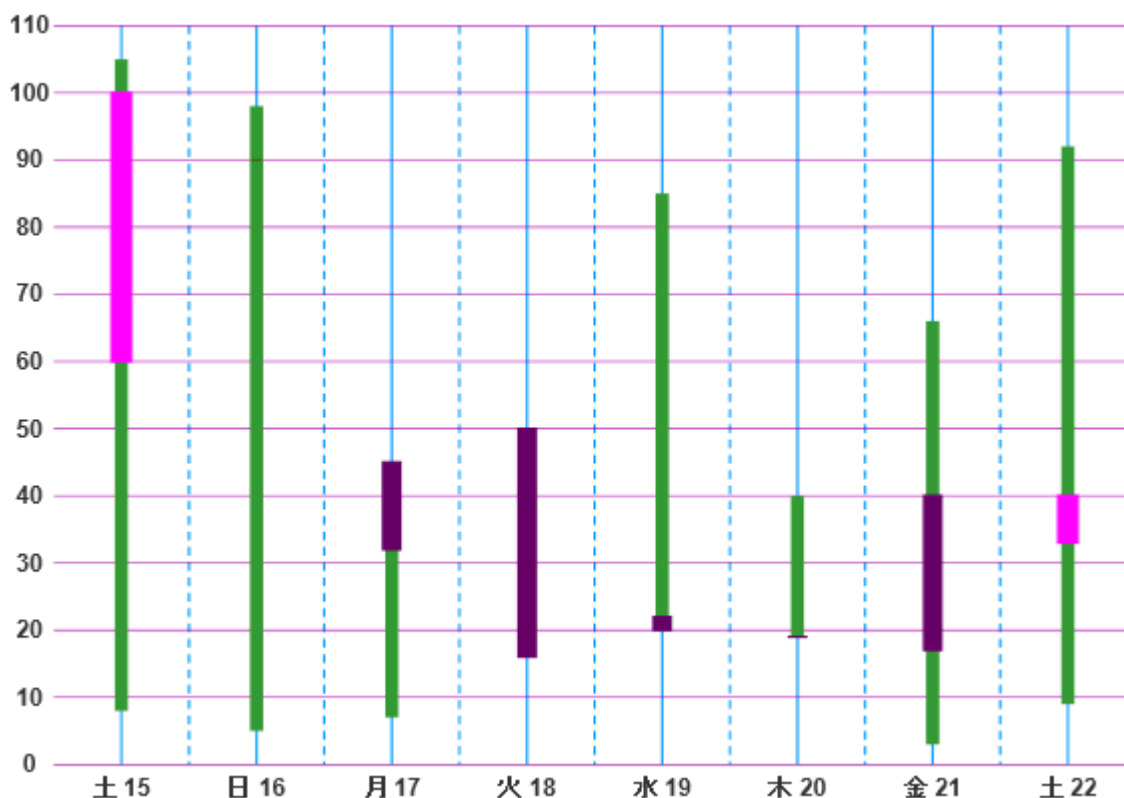
  <Y Visible="False" Compass="West">
    <Labels TextAlign="Center"></Labels>

    <GridMajor Visible="True">
      <Style Stroke="#B300B3"></Style>

      <GridStyle Stroke="#B300B3"></GridStyle>
    </GridMajor>

    <GridMinor Visible="False"></GridMinor>
  </Y>
</Axis>
```

CandlestickChart for ASP.NET Web Forms



軸の範囲

グラフには通常、そのグラフに含まれているすべてのデータが表示されます。ただし、軸の範囲を固定することで、チャートの特定の部分を表示できます。

各軸の範囲は、データの最小値/最大値と目盛りの間隔を考慮して決定されます。**Axis.Min**、**Axis.Max**、**Axis.AutoMin**、**Axis.AutoMax** の各プロパティを設定すると、このプロセスをカスタマイズできます。

● 軸の最大値と最小値

チャートを特定の軸範囲に限定するには、**Axis.Min** プロパティと **Axis.Max** プロパティを使用します。チャートの X 軸の値の範囲が 0 ~ 100 の場合に、**Axis.Min** を 0 に、**Axis.Max** を 10 に設定すると、10 までの値だけが表示されます。

チャートの **Axis.Min** と **Axis.Max** の値を自動的に計算することもできます。**Axis.AutoMax** プロパティと **Axis.AutoMin** プロパティを **True** に設定すると、現在のデータセットに合わせて軸の数値目盛りが自動的に書式設定されます。

軸の注釈

各軸の注釈は、どのようなチャートでも重要な部分です。チャートデータが変わった場合でも、できるだけ適切な注釈が自動的に作成されます。

次のプロパティは、**C1CandlestickChart** の軸の注釈の書式とレイアウトを表します。

軸プロパティ	説明
--------	----

ChartAxis.AnnoFormatString	注釈の書式文字列を示す値。
ChartAxis.AnnoMethod	注釈のメソッドを示す値。
ChartAxis.ValueLabels	軸の valueLabel のコレクションを示す値。

数値の注釈

数値の注釈を実装すると、チャートのデータ自体から数値の注釈が自動的に生成されます。数値の注釈は、任意の軸、グラフタイプ、およびデータレイアウトに使用できます。数値の注釈は、軸の次のプロパティによって制御されます。

プロパティ	説明
ChartAxis.AnnoFormatString	注釈の書式文字列を示す値。

ヘッダーとフッター

ヘッダー要素とフッター要素は、グラフの説明を表示するために使用されます。これらは、**Header** プロパティと **Footer** プロパティで制御されます。

グラフのヘッダープロパティとフッタープロパティは、次の主要プロパティを含む **ChartTitle** オブジェクトを返します。

プロパティ	説明
ChartTitle.Compass	タイトルの方角を示す値。これは、タイトルの位置を North(グラフの上)、South(グラフの下)、East(グラフの右)、West(グラフの左)で指定します。
ChartTitle.Style	タイトルのフォント、方向、色、および境界線を設定するプロパティを含みます。
ChartTitle.Text	タイトルの位置を決定します。
ChartTitle.TextStyle	タイトルテキストのスタイルを示す値。
ChartTitle.Visible	タイトルが表示されているかどうかを決定します。

C1CandlestickChart は、コンテンツと **ChartTitle.Compass** プロパティの設定に基づいて、サイズと位置を自動的に調整します。

- **ヘッダー要素とフッター要素のカスタマイズ**

ヘッダー要素とフッター要素のテキスト、配置、位置、境界線、色、およびフォントは、**ChartTitle** のプロパティを使用してカスタマイズできます。

ヘッダー要素は、**X** プロパティを使用して左(負の値)または右(正の値)に移動したり、**Y** プロパティを使用して上(正の値)または下(負の値)に移動することができます。

凡例

凡例要素は、チャートの各データ系列に関する情報を表示します。チャート凡例は、物理色とデータ系列の間のマッピングを表します。

C1CandlestickChart は、グラフにデータが存在し、**ChartSeries.LegendEntry** プロパティが有効な場合は常に**凡例**を自動生成します。**ChartSeries.Label** プロパティを指定すると、各系列の名前が凡例に表されます。**ChartSeries.Label** プロパティに値が指定されていない場合、系列の名前は、未定義として凡例に表示されます。

凡例は、**ChartLegend** プロパティによって制御されます。このプロパティは、次の主要プロパティを含む **ChartLegend** オブジェクトを返します。

プロパティ	説明
ChartLegend.Text	凡例のタイトルに表示されるテキストを含みます。
ChartLegend.Style	凡例のフォント、方向、色、および境界線を設定するプロパティを含みます。
ChartLegend.Compass	凡例の位置を決定します。
ChartLegend.Visible	凡例が表示されているかどうかを決定します。
ChartLegend.Orientation	凡例項目を水平方向と垂直方向のどちらの方向に並べて表示するかを決定します。

C1CandlestickChart は、凡例のコンテンツと **ChartLegend.Compass** プロパティおよび **ChartLegend.Orientation** プロパティに基づいて、凡例のサイズと位置を自動決定します。

凡例の方向は、**ChartLegend.Orientation** プロパティを使用して、水平方向または垂直方向にすることができます。凡例の位置は、**ChartLegend.Compass** プロパティを使用して、東西南北のいずれかに設定できます。

系列

CandlestickChartSeries オブジェクトは、ローソク足チャートにプロットされるデータを表します。CandlestickChartSeries をデザインビューで追加するには **CandlestickChartSeries コレクションエディタ**、ソースビューで追加するには CandlestickChartSeries 要素、プログラムで追加するには CandlestickChartSeries オブジェクトをそれぞれ使用します。

- **CandlestickChartSeries をデザインビューで追加するには**
 1. **C1CandlestickChart** コントロールを選択し、スマートタグをクリックします。
 2. **[C1CandlestickChart のタスク]**メニューで **SeriesList** 項目を選択して、**[CandlestickChartSeries コレクションエディタ]**ダイアログボックスを開きます。
 3. **[追加]**をクリックして、**SeriesList** コレクションに **CandlestickChartSeries** メンバを追加します。
- **CandlestickChartSeries をソースビューで追加するには**

ソースビュー

```
<cc1:CandlestickChartSeries Label="8 Days" LegendEntry="True">
  <Data>
    <High DoubleValues="105, 98, 45, 50, 85, 40, 66, 92"></High>

    <Low DoubleValues="8, 5, 7, 16, 20, 19, 3, 9"></Low>

    <Open DoubleValues="100, 90, 32, 16, 20, 8, 17, 40"></Open>

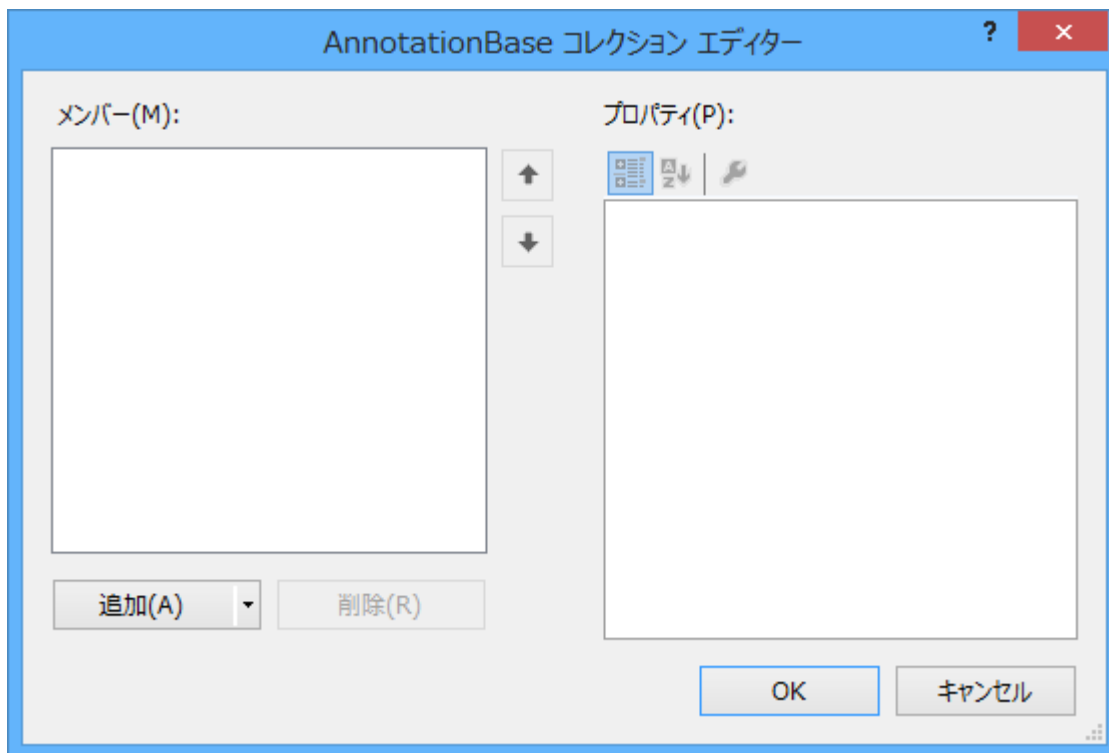
    <Close DoubleValues="60, 90, 45, 50, 22, 19, 40, 33"></Close>

    <X>
      <Values>
        <cc1:ChartXData DateTimeValue="2014-03-15"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-16"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-17"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-18"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-19"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-20"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-21"></cc1:ChartXData>
        <cc1:ChartXData DateTimeValue="2014-03-22"></cc1:ChartXData>
      </Values>
    </X>
  </Data>
</cc1:CandlestickChartSeries>
```

注釈

C1BarChartでは、**Annotations** プロパティを設定して注釈を追加できます。注釈を追加またはカスタマイズするために AnnotationBaseコレクションエディタを使用することができます。

CandlestickChart for ASP.NET Web Forms

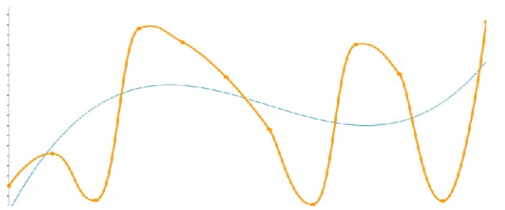
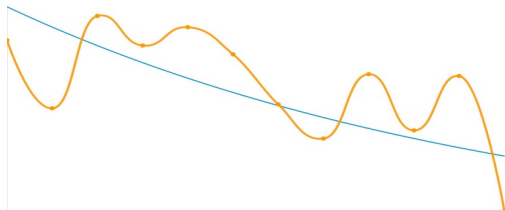
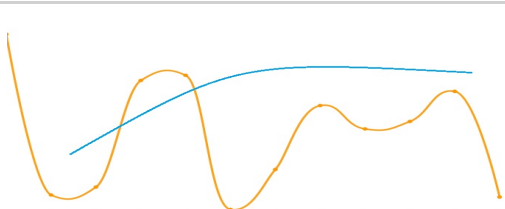
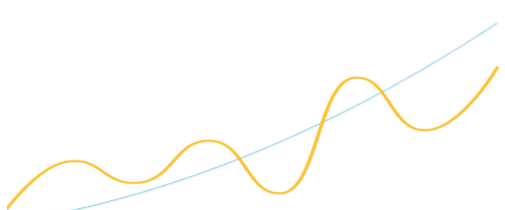
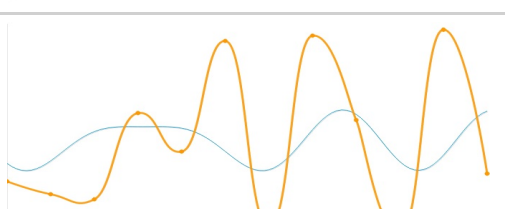
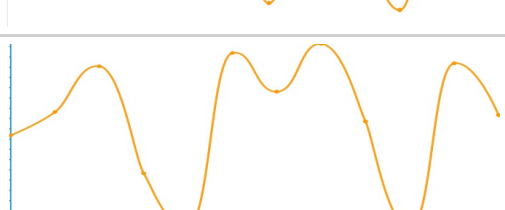


近似曲線

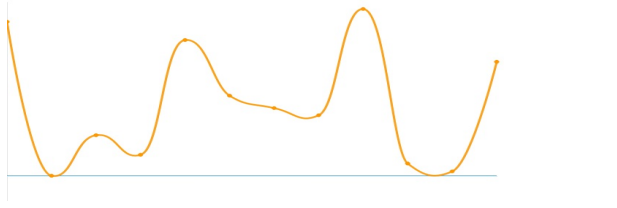
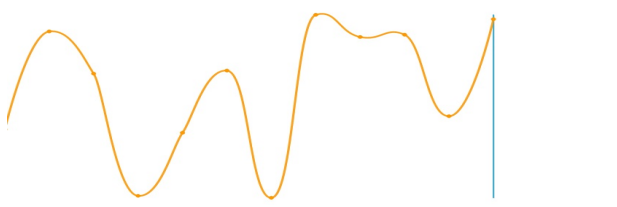
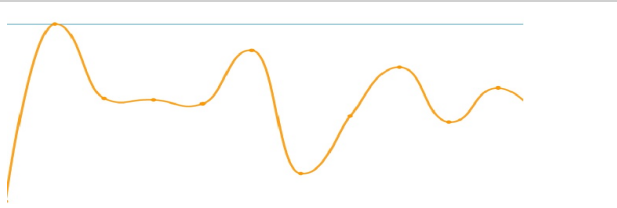
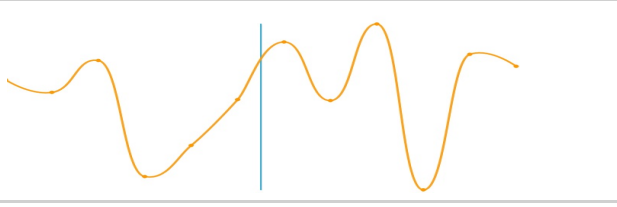
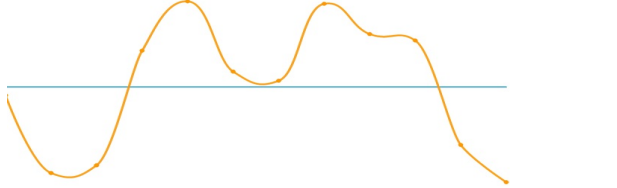
近似曲線の追加近似曲線は、データの傾向を示し、予測問題の検討に利用されるツールです。価格チャートや金融チャートで使用されるのが多いですが、株価のテクニカル分析で使用される取引指標、MACD(Moving Average Convergence Divergence、移動平均収束拡散法)や金融市場の解析に使用されるテクニカル指標、RSI(Relative Strength Index、相対力指数)のようなテクニカル分析グラフでも使用されます。

近似曲線の種類

以下の一覧で対応するすべてのFitTypesが説明されています。近似曲線の各種類がそれぞれの計算式に基づいて作成されます。

FitType	説明	例画像
Polynom	データが変動する場合に使用される曲線です。大量のデータで利益や損失を分析するのに採用されます。	
Exponent	データ値の増減率が高くなっていく場合に適します。0 または負の値が含まれているデータの場合、指数近似曲線を作成できません。	
Logarithmic	データの可視化に最適する曲線です。データの変化率が急速に増加または減少してから平になる場合に適します。この曲線では負の値と正の値を使用できます。	
Power	特定の比率で増加する測定を比較するデータセットの場合に適します。たとえば、超特急列車の加速を1秒間隔で計測した値など。	
Fourier	フーリエ級数式で作成され、波動関数を簡単な正弦波の組み合わせとして表示する方法です。	
Min X	グラフからXの最小値を取得して近似曲線を作成します。	

CandlestickChart for ASP.NET Web Forms

Min Y	>グラフからYの最小値を取得して近似曲線を作成します。	
Max X	グラフからXの最大値を取得して近似曲線を作成します	
Max Y	グラフからYの最大値を取得して近似曲線を作成します。	
Average X	グラフからXの平均値を取得して近似曲線を作成します。	
Average Y	グラフからYの平均値を取得して近似曲線を作成します。	

近似曲線をデザイナーまたは、ソースビューやコードでも追加できます。C1BarChartに近似曲線を追加するには、以下の手順に従います。

デザイナーの場合

以下の手順では、チャートにデータが追加されていることを前提します。詳細については、C1BarChartの[クイックスタート](#)を参照してください。

1. **CandlestickChart** スマートタグをクリックし、[**C1CandlestickChart タスク**]メニューから **シリーズ一覧** を選択します。**CandlestickChartSeries コレクションエディター**が表示されます。
2. **CandlestickChartSeriesコレクションエディター**で、**<追加>**ボタンをクリックして、新しい **CandlestickChartSeries** を BarChartに追加します。
3. 新規に追加された系列の**isTrendline**プロパティをTrueに設定します。
4. **Trendline Series**のプロパティグループを展開して以下のプロパティを設定できます。

▲ TrendlineSeries	C1.Web.Wijmo.Control
▷ Data	C1.Web.Wijmo.Contro
FitType	MaxY
Order	4
SampleCount	150

- **FitType** -近似曲線の種類を指定します。
- **SampleCount**-近似曲線の関数計算用にサンプルカウントを指定します。FitTypeがpolynom、power、exponent、logarithmicやfourierの場合のみ機能します。

- **Order** - 多項式の次数を指定します。FitTypeがpolynom、power、exponent、logarithmicやfourierの場合のみ機能します。
5. TrendlineSeries.DataのプロパティグループにあるTrendlineSeries.Data.X.Valuesプロパティの横にある<...>ボタンをクリックして**ChartXData コレクションエディター**を開きます。
 6. X軸に表示される値を入力して、**OK**をクリックします。
 7. TrendlineSeries.DataのプロパティグループにあるTrendlineSeries.Data.Y.Valuesプロパティの横にある<...>ボタンをクリックして**ChartYData コレクションエディター**を開きます。
 8. Y軸に表示される値を入力して、**OK**をクリックします。
 9. **FitType**プロパティ、**Order**プロパティと**SampleCount**プロパティを設定します。
 10. **<OK>**をクリックして保存し、**CandlestickChartSeriesコレクションエディター**を閉じます。

ソースビューの場合

<SeriesList></SeriesList>タグ内に以下のマークアップを追加してグラフに近似曲線を追加します。

SourceView

```
<ccl:CandlestickChartSeries LegendEntry="True" IsTrendline="true">

  <TrendlineSeries FitType="Polynom">
    <Data>
      <X>
        <Values>
          <ccl:ChartXData StringValue="QTR1" />
          <ccl:ChartXData StringValue="QTR2" />
          <ccl:ChartXData StringValue="QTR3" />
          <ccl:ChartXData StringValue="QTR4" />
        </Values>
      </X>
      <Y>
        <Values>
          <ccl:ChartYData DoubleValue="13" />
          <ccl:ChartYData DoubleValue="4" />
          <ccl:ChartYData DoubleValue="18" />
          <ccl:ChartYData DoubleValue="8" />
        </Values>
      </Y>
    </Data>
  </TrendlineSeries>

</ccl:CandlestickChartSeries>
```

コードの場合

Page_Loadイベントに以下のコードを追加することでチャートに近似曲線を追加できます。

C#コードの書き方

C#

```
// 新規系列を作成します。
var seriesTrendline = new CandlestickChartSeries();
```

CandlestickChart for ASP.NET Web Forms

```
seriesTrendline.IsTrendline = true;
seriesTrendline.Label = "Trendline";
seriesTrendline.TrendlineSeries.FitType = TrendlineFitType.Polynom;
seriesTrendline.TrendlineSeries.Order = 4;
seriesTrendline.TrendlineSeries.SampleCount = 100;

// チャートに系列を追加します。
this.C1BarChart1.SeriesList.Add(seriesTrendline);

// X軸のデータを追加します。
seriesTrendline.TrendlineSeries.Data.X.Add("QTR1");
seriesTrendline.TrendlineSeries.Data.X.Add("QTR2");
seriesTrendline.TrendlineSeries.Data.X.Add("QTR3");
seriesTrendline.TrendlineSeries.Data.X.Add("QTR4");

// Y軸のデータを追加します。
seriesTrendline.TrendlineSeries.Data.Y.Add(13);
seriesTrendline.TrendlineSeries.Data.Y.Add(4);
seriesTrendline.TrendlineSeries.Data.Y.Add(18);
seriesTrendline.TrendlineSeries.Data.Y.Add(8);
```

Visual Basicコードの書き方

VB

```
' 新規系列を作成します。
Dim seriesTrendline = New CandlestickChartSeries()
seriesTrendline.IsTrendline = True
seriesTrendline.Label = "Trendline"
seriesTrendline.TrendlineSeries.FitType = TrendlineFitType.Polynom
seriesTrendline.TrendlineSeries.Order = 4
seriesTrendline.TrendlineSeries.SampleCount = 100

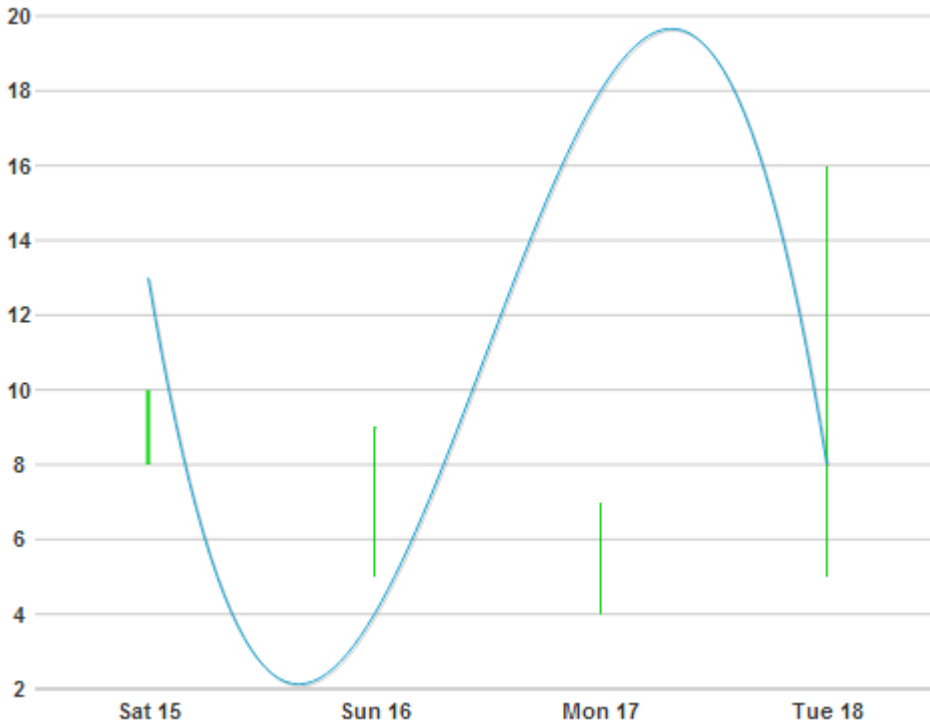
' チャートに系列を追加します。
Me.C1BarChart1.SeriesList.Add(seriesTrendline)

' X軸のデータを追加します。
seriesTrendline.TrendlineSeries.Data.X.Add("QTR1")
seriesTrendline.TrendlineSeries.Data.X.Add("QTR2")
seriesTrendline.TrendlineSeries.Data.X.Add("QTR3")
seriesTrendline.TrendlineSeries.Data.X.Add("QTR4")

' Y軸のデータを追加します。
seriesTrendline.TrendlineSeries.Data.Y.Add(13)
seriesTrendline.TrendlineSeries.Data.Y.Add(4)
seriesTrendline.TrendlineSeries.Data.Y.Add(18)
seriesTrendline.TrendlineSeries.Data.Y.Add(8)
```

このトピックの作業結果

プロジェクトを実行すると、チャートに緑色の近似曲線が表示されます。



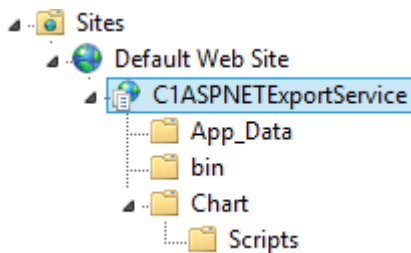
エクスポートサービス

C1 ASP.NET Export Service を使用すると、複雑なエクスポートアプリケーションを作成しなくても、チャートを画像または PDF としてエクスポートできます。このサービスはアプリケーションサーバー上に置かれます。

C1 ASP.NET Export Service を使用する利点は次のとおりです。

- 書式設定を維持したままチャートをエクスポートできます。
- エクスポートされたファイルの設定を必要に応じて変更できます。

Export Service は、インターネットインフォメーションサービス (IIS) 上に配布される Web アプリケーションです。C:\Program Files\ComponentOne\ASP.NET Web Forms フォルダにある **C1ASPNETExportService** インストーラを実行してください。次のファイルが IIS にインストールされます。




これらのファイルは、次の場所にも格納されています。

C:\ProgramData\ComponentOne\C1ASPNET\C1APNETExportService

システム要件

サービスホストのシステム要件は次のとおりです。

- Microsoft Windows 7 以上。
- ASP.NET 4.0 以上 (.NET Framework 4.0) を含む IIS 7.0 以上。
- サービスホスト上に IE9 以上。

 **Microsoft Windows 7** または **Microsoft Windows Server 2008 R2** では .Net Framework 4.0 をアップデートできます。詳細については、「<http://support.microsoft.com/kb/2468871>」を参照してください。

チャートのエクスポート設定

チャートをエクスポートするための設定は次のとおりです。

画像としてエクスポート

- **ファイル形式**: .jpg、.bmp、.gif、.png、または .tiff 画像としてエクスポートします。
- **作成者**: データの作成に対して責任を負う人または組織の名前を指定します。
- **サーバー URL**: サーバーの URL を設定します。"<サーバーの URL>/exportapi/chart" を入力します。
- **ファイル名**: エクスポートされる画像に使用するファイル名を設定します。

PDF にエクスポート

- **幅の自動調整**: 自動調整を有効にします。
- **横**: 横長モードを有効にします。
- **ファイルの内容**:
 - 画質: 画質を Low (低)、Medium (中)、または High (高) に設定します。
 - 圧縮: 圧縮レベルを Default (デフォルト)、None (なし)、Best Speed (速度優先)、または Best Compression (圧縮率優先) に設定します。
 - フォントタイプ: フォントタイプを True Type または Embedded (埋め込み) に設定します。

トピックの内容

システム要件

チャートのエクスポート設定 用途

● ドキュメント情報:

- 作成者:ドキュメントを作成した人または組織の名前を設定します。
- 作成元アプリケーション:元のドキュメントを作成したアプリケーションの名前を設定します。
- サブタイトル:ドキュメントのサブタイトルを設定します。
- タイトル:タイトルバーに表示されるドキュメントのタイトルを設定します。
- PDF 作成アプリケーション:PDFドキュメントを作成したアプリケーションの名前を設定します。
- キーワード:PDFドキュメントに関連付けられるキーワードを設定します。これらのキーワードは、ドキュメントの検索に使用できます。

● ドキュメントのセキュリティ:

- 暗号化の種類:暗号化の種類を NotPermit(許可しない)、Standard40、Standard128、または Aes128 に設定します。
- 所有者パスワード:ドキュメントの権限を編集するために必要なパスワードを設定します。
- ユーザーパスワード:ドキュメントを開くために必要なパスワードを設定します。
- コンテンツのコピーを許可:コンテンツのコピーを許可または禁止します。
- 注釈の編集を許可:ユーザーが注釈を編集することを許可または禁止します。
- コンテンツの編集を許可:ユーザーがドキュメントの内容を編集することを許可または禁止します。
- 印刷を許可:ドキュメントの印刷を許可または禁止します。

● 構成設定:

- サーバー URL:サーバーの URL を設定します。"<サーバーの URL>/exportapi/chart" を入力します。
- ファイル名:エクスポートされる PDF に使用するファイル名を設定します。

用途

チャートを画像または PDF としてエクスポートするには、`exportChart` メソッドを呼び出します。チャートを画像にエクスポートするには、`<head>` タグと `</head>` タグの間に次のコードを追加します。

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js" type="text/javascript">
</script>
<asp:PlaceHolder runat="server">
  <!--Export メソッド-->
  <script type="text/javascript">
    $(function () {
      $("#Button1").click(exportImage);
    });
    function getChart() { return $("#<%=C1CandlestickChart1.ClientID%>"); }

// Export 関数
function exportImage() {
var fileName = "ExportImage";
var type = "Png";
var url = "http://demos.componentone.com/ASPNET/ExportService" + "/exportapi/chart";
var chart = getChart();
$("#<%=C1CandlestickChart1.ClientID%>").c1candlestickchart("exportChart", fileName,
  type, url);
}
</script>
</asp:PlaceHolder>
```

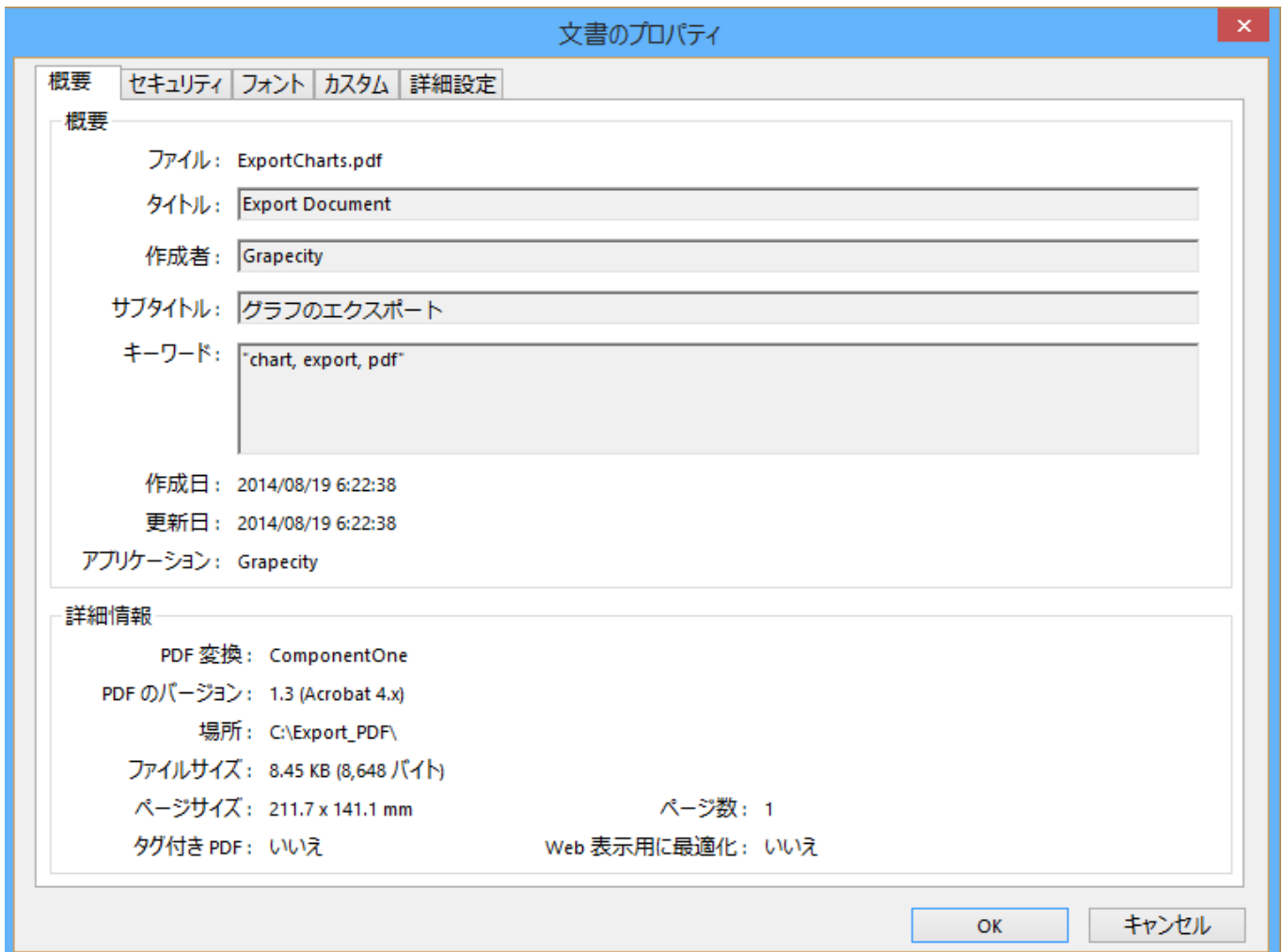
チャートを PDF にエクスポートするには、`<head>` タグと `</head>` タグの間に次のコードを追加します。

```
<script src="http://code.jquery.com/jquery-1.9.1.min.js" type="text/javascript">
</script>
<asp:PlaceHolder runat="server">
<script type="text/javascript">
  $(function () {
    $("#Button1").click(exportPdf);
  });
</script>
```

CandlestickChart for ASP.NET Web Forms

```
});  
  
// Export function  
function exportPdf() {  
var fileName = "ExportCharts";  
var url = "http://demos.componentone.com/ASPNET/ExportService" + "/exportapi/chart";  
var pdfSetting = {  
  imageQuality: 'Low',  
  compression: 'BestCompression',  
  fontType: 'TrueType',  
  author: 'Grapecity',  
  creator: 'Grapecity',  
  subject: 'グラフのエクスポート',  
  keywords: 'chart, export, pdf',  
  allowCopyContent: true,  
  allowEditAnnotations: true,  
  allowEditContent: true,  
  allowPrint: true  
}  
  
$("#<%=C1CandlestickChart1.ClientID%>").c1candlestickchart("exportChart", fileName,  
  "pdf", pdfSetting, url);  
}  
</script>  
</asp:Placeholder>
```

生成される PDF のプロパティを次の図に示します。



ファイルが Internet Explorer でダウンロードされない場合は、Internet Explorer の保護モードをオフにしてファイルをエクスポートするか、Internet Explorer を管理者として実行します。保護モードをオフにするには、次のようにします。

- Internet Explorer の [設定]を開き、[インターネットオプション]を選択します。
- [セキュリティ]タブで、[インターネット]を選択し、[保護モードを有効にする]のチェックを外します。