How to Use ChartControls

**Introduction**

This article describes how to use ChartControls control.

ChartControls is used to display the historical trend of the stock, time trends and movements in real-time graphical controls.

**Supported Platforms**

Classic Windows desktop

Windows 8

Windows Phone 8

Universal Windows Platform

**Prerequisite**

Visual Studio 2012 or any other version.

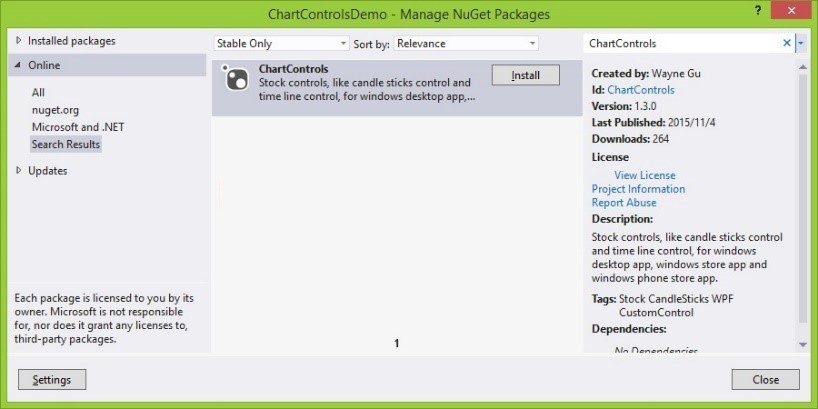
Net 4.0 or later.

Nuget.

Familiar with C # and WPF.

**Get ChartControls**

In Visual Studio, with Nuget search "ChartControls" or "Stock", find ChartControls control to download and install.



**Create a price curve**

1. Use Visual Studio 2012 to create a WPF project.
2. Download and install ChartControls Nuget control.
3. Edit MainWindow.xaml

Add References

xmlns: local = "clr-namespace: ChartControls; assembly = ChartControls"

Add Controls

<Local: ChartControl Name = "priceControl" Margin = "0,5,0,0"> </ local: ChartControl>

1. Edit MainWindow.xaml.cs

Import Namespace

using ChartControls;

using ChartControls.Drawing;

1. Code snippets

DataLoader loader;

public MainWindow ()

{

    InitializeComponent ();

    loader = new DataLoader ("s.json", true);

    CreateCurve ();

}

public void CreateCurve ()

{

    string id = "000001";

    // Load chart items

    IList <ChartItem> chartItems = loader.GetChartItems (id);

    // Create collection id

    CollectionId collId = new CollectionId (id);

    // Create pen

    IPen pen = DrawingObjectFactory.CreatePen (Brushes.Black, 1);

    // Create chart item collection

    ChartItemCollection collection = new ChartItemCollection (collId, chartItems, pen, null);

    // Set main collection

    priceControl.SetMainCollection (collection);

}

1. Screenshot



**Create Candle stick line graph**

1. See the preceding step on one example.
2. Edit MainWindow.xaml

Add References

xmlns: local = "clr-namespace: ChartControls; assembly = ChartControls"

Add Controls

<Local: ChartControl Name = "priceControl" Margin = "0,5,0,0"> </ local: ChartControl>

<Local: ChartControl Name = "VolumnControl" Grid.Row = "1" YCursorFormat = "N0" YScaleFormat = "N0"> </ local: ChartControl>

1. Code snippets

public void CreateCandle ()

{

    string id = "000001";

    // Load chart items

    StockVolumnList svList = loader.GetStockItems (id);

    // Create collection id

    CollectionId collId = new CollectionId (id);

    // Create pens

    IPen raisePen = DrawingObjectFactory.CreatePen (Brushes.Red, 1);

    IPen fallPen = DrawingObjectFactory.CreatePen (Brushes.Green, 1);

    // Create stock item collection

    StockItemCollection stockColl = new StockItemCollection (collId, svList.Prices, raisePen, fallPen, null);

    // Set main collection

    priceControl.SetMainCollection (stockColl);

    // Create volumn item collection

    VolumnItemCollection volumnColl = new VolumnItemCollection (collId, svList.Volumns, raisePen, fallPen);

    // Set main collection

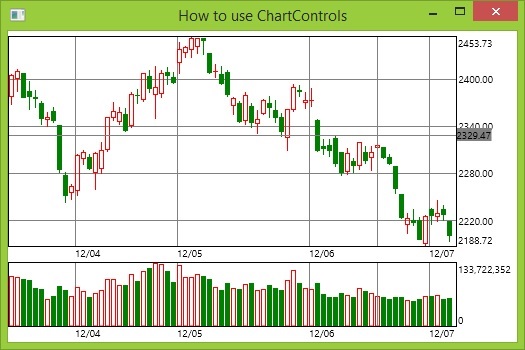
    volumnControl.SetMainCollection (volumnColl);

    // Connect two controls

    priceControl.AddConnection (volumnControl);

}

1. Screenshot



**Creating time graph**

1. See the first example in the previous step.
2. Add controls

<Local: ChartControl Name = "priceControl" Margin = "0,5,0,0" XScaleFormat = "HH: ss"> </ local: ChartControl>

<Local: ChartControl Name = "VolumnControl" Grid.Row = "1" YCursorFormat = "N0" YScaleFormat = "N0" XScaleFormat = "HH: ss"> </ local: ChartControl>

1. Code snippets

public void CreateTime ()

{

    string id = "600100";

    StockVolumnList svList = timeLoader.GetStockItems (id);

    // Create collection id

    CollectionId collId = new CollectionId (id);

    // Create pens

    IPen raisePen = DrawingObjectFactory.CreatePen (Brushes.Red, 1);

    IPen fallPen = DrawingObjectFactory.CreatePen (Brushes.Green, 1);

    // Create stock item collection

    SymmetricChartItemCollection stockColl = new SymmetricChartItemCollection (collId, svList.Prices, raisePen, null, SymmetricCommonSettings.CNSettings);

     // Set main collection

     priceControl.SetMainCollection (stockColl);

     // Create volumn item collection

     SymmetricVolumnItemCollection volumnColl = new SymmetricVolumnItemCollection (collId, svList.Volumns, raisePen, fallPen, SymmetricCommonSettings.CNSettings);

     // Set main collection

     volumnControl.SetMainCollection (volumnColl);

     // Connect two controls

     priceControl.AddConnection (volumnControl);

 }

1. Screenshot



**Other Supported Graphics**

Multi-map overlay, multi-map merge, dynamic map, custom graphics.

**Property List**

Controls have many attributes, you can set, such as background color, border color and width, font and so on.

**Demonstration program address**

https://github.com/Waynext/ChartControls/tree/master/Src/Tests

The demonstration program contains multiple platforms.

**Summary**

ChartControls is an easy to use stock control, with the freedom to adjust the controls look through control properties. You can also extend more customized graphics by inheritance. Control supports all Windows platforms.