如何使用ChartControls

**概论**

本文主要介绍如何在代码中使用ChartControls控件。

ChartControls是用于显示股票历史走势，分时走势及实时走势的图形控件。

**支持平台**

Classic Windows desktop

Windows 8

Windows Phone 8

Universal Windows Platform

**使用条件**

Visual Studio 2012或者其他版本。

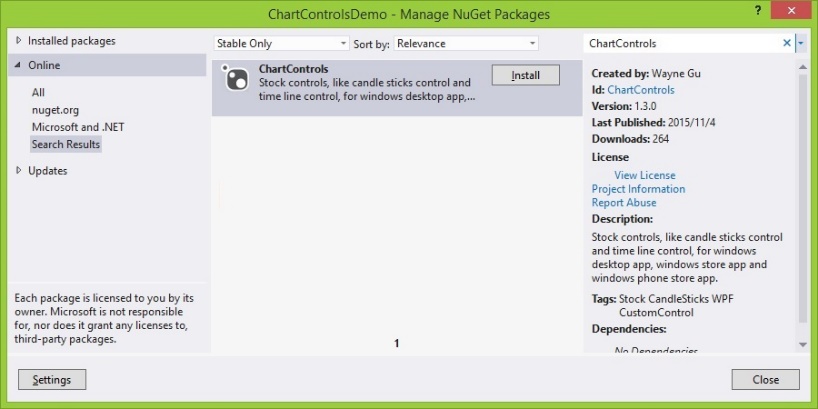
Net 4.0 或者更高版本。

Nuget。

熟悉C# 和 WPF。

**获取ChartControls**

在Visual Studio中，用Nuget搜索”ChartControls” 或者 “Stock”，找到ChartControls控件下载安装。



**创建价格曲线**

1. 用Visual Studio 2012创建WPF项目。
2. 用Nuget下载并安装ChartControls控件。
3. 编辑MainWindow.xaml

添加引用

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

添加控件

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

1. 编辑MainWindow.xaml.cs

导入命名空间

using ChartControls;

using ChartControls.Drawing;

代码片段

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. 截图



**创建K线图**

1. 前面步骤参见上一个例子。
2. 编辑MainWindow.xaml

添加引用

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

添加控件

<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. 代码片段

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. 截图



**创建分时图**

1. 前面步骤参见第一个例子。
2. 添加控件

<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. 代码片段

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. 截图



**其他支持的图形**

多图叠加，多图合并，动态图，自定义图形。

**属性列表**

控件有很多属性，可以根据需要设置，比如背景颜色，边框颜色及宽度，字体等等。

**完整演示程序地址**

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

包含多个平台的演示程序。

**总结**

ChartControls是一款简单易用的股票控件，可以通过控件属性自由调整控件外观。也可以通过继承的方式，扩展更多自定义的图形。控件支持所有Windows平台。