# Requirements

* Build a C# app that downloads and display historical price data from Yahoo Finance.
  + The minimal set of data should have open, high, low and closing prices
  + Research the Yahooo! website and see how you can grab this data
  + User should be able to select the ticker and range of dates. Ie. 0005.HK Between Jan 1, 2013 to May 1, 2014
* It should also have the capability to download and display headlines from RSS feed
  + <http://feeds.finance.yahoo.com/rss/2.0/headline?s=0005.HK&region=US&lang=en-US>
* Calculate and display 30 day volatility and 30 day moving averages
* Design a simple interface to show the data

# Development

Development environment: Visual Studio 2010 + .NET 4.0

Pass the test under Windows 7 and Windows XP.

Third party libraries:

* Syncfusion

<http://www.syncfusion.com/products/wpf>

Mainly use its Docking manager.

I use the latest trial version, and can run the compiled program from our local machine. Not sure whether needs to install this trial version if you want to compile my codes. Maybe there are some limitations to compilation.

* Amchars

It is an open-source library, and the link is <https://github.com/ailon/amCharts-Quick-Charts>

Mainly use its candle chart function.

You can directly add the “AmCharts.Windows.Stock.dll” from my project and compile it.

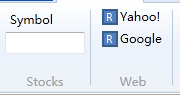
# Functionalities

## Overview



## Fetch Data

* Fetch from Yahoo Finance



Enter your wanted stock symbol (case insensitive), and then click “Yahoo!” button, it will download all historical data (from 1962), and then save it into local CSV file into the folder “data/” under the exe directory. As well as it will add one list item into “Stock Overview” listbox.

If you already downloaded it before, it will overwrite the CSV file and just update the listbox item.

* Fetch from local CSV file

Every time the program starts, it will automatically loads all the data file under the “data/” folder.

## Chart



This panel contains multi-charts, and each tab represents one stock chart. The top-downs chars are:

* Candle chart: show the open/close/high/low price for each ticker, and show moving average price within the same chart to convenient comparing the prices.
* Volume chart: show the volume
* Volatility chart: show the volatility line
* Axes chart: we can zoom-in and zoom-out use this chart.

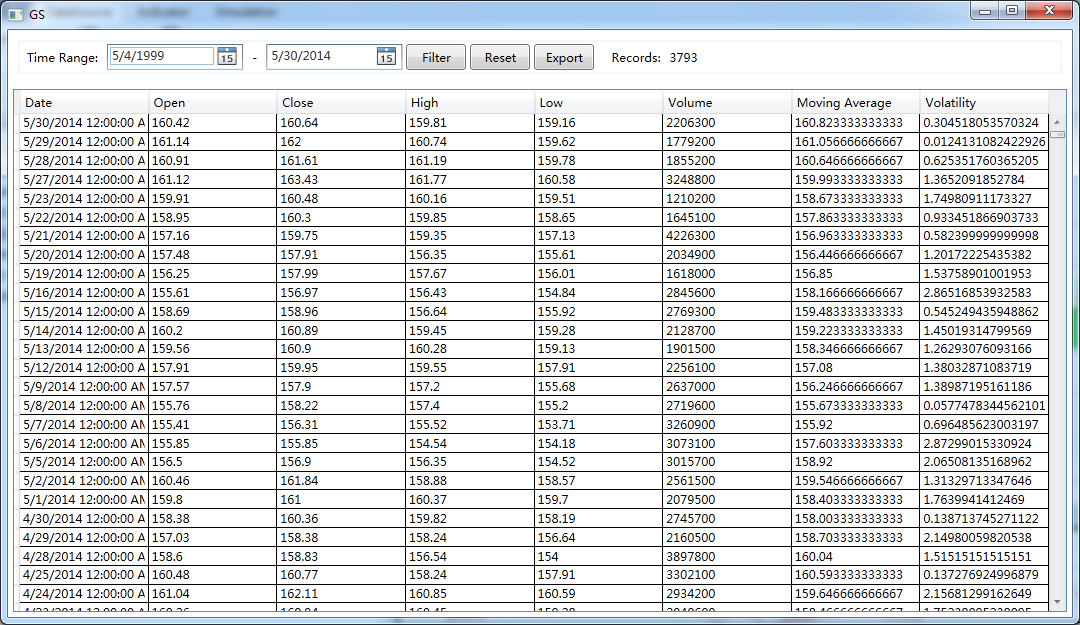
You can use “custom period” to filter the specific time range data, and it happens when the two text boxes lose focus. Also you can use pre-defined periods to filter the data.

The all charts will show the data value simultaneously according to your cursor position, and they are sharing the same timestamp.

NOTE:

1. The chart plug-in will automatically zoom-in and out to the different time span. Please notify the datetime displayed in the right-upper corner.
2. The open-source plug-in has some bugs to show the candle color. If the ticker is week or month, it works fine. But if the ticker is day, all the candle will be green.
3. The chart will show after you click the button “Chart”, and its performance is really bad. Please select the targeted stock in the “stock overview” first to active the “Chart” button, and be patient when there is some stuck of the program. Originally I had plan to add the BusyIndicator, but I have no enough time to achieve this.

## Grid

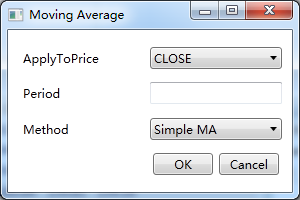


It will display all the ticker data and allows you to filter the data using the “Filter” button. You can revert the whole data through “Reset”. The “Export” is design to save the filtered results into excel, but have not implemented.

NOTE:

1. The performance of the grid is not so good as the data volume is large. It may stuck for a while after you click “Grid” button. Please be patient to wait its finish.

## Indicator

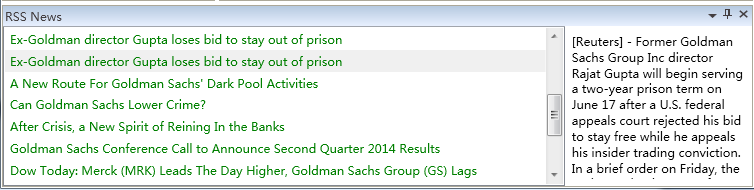


You can adjust the indicator settings by click “Indicator” group and press “OK” to retrigger the indicator calculation. It will reccalculate related indicator values with given stock.

NOTE:

1. The values in the grid will automatically update after your retrigger the calculation. But the chart will not update due to the Amcharts plug-in bugs.

## RSS



The program will retrieve the RSS news with the given stock from Yahoo RSS interface. The “RSS News” panel will hide in the left-buttom side when the program shows.

You can select the different titles and the right textbox will automatically display its description and link address.

# Codes

The code design according to the MVVM pattern which encouraged by Microsoft WPF.

It contains the following folders:

1. Model: basic information about stock and rss
2. ViewModel: the changed-notification wrapper for the model
3. View: the related GUI part
4. Loader: the loaders to retrieve the data from different data sources
5. Indicator: simple indicator implementation

# Future Improvements

* Chart and Grid performance

With the limitation of the open source project, the performance and stability are not quite good. It took me a lot of time to fit my data into this kind of chart.

I would wish to add the busy indicator to increase the user experience, but it should import another 3rd party library call “ExtentedTookKit”. Please be patient after you click “Chart” and “Grid”. Sorry for the inconvenient.

Originally I design that every time you switch the stock in the list, it will automatically load the chart. But I found this is very risky to miss click, and make you have to wait there. So I improve it by you have to explicitly click the “Chart” button for the first time loading. After the chart already shows in the tabs, it will be automatically switched and displayed in the front GUI when you change the stocks.

* Indicator integration

The indicator integration is not natural, and the chart will not update after you refresh the indicator. The Amchars toolkit is not so user friendly, and it lacks documents.

The chart does not support to bind multiple data sources, so I took a not elegant way to hack it. You can see in my codes there are three hard-coding datasets in it which makes the future expansion a little difficult.

Also you maybe curious that why I design the data-structure of StockInfo, and mangling the indicator’s value into it. This is also limited by the chart data binding. Ideally I will separate the indicator values from the basic information, so that is the reason I write the calculate2 function in MovingAverageIndicator.

* Others

I have tried my best to make this program bug-free and user friendly. Due to the time limitation, it definitely contains some unknown bugs.If you find some bugs or inconvenient functionalities, please tell me and I will improve it later.

Thanks.