FOREX Trading and Investment

An Interactive Qualifying Project Report

Submitted to the Faculty

of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the

Degree of Bachelor of Science

By

Mariela Qirici

Sebastian Franco

Jonathan Baiden

Craig Nesbitt

Approved by: Prof. Hossein Hakim

November 11, 2013

Abstract

This Interactive Qualifying Project serves as an introduction to the world of Foreign Exchange trading by giving a basic guideline as to how one can get started with FOREX trading and the development of a trading strategy. It concentrates on designing a Trading Strategy that if profitable, can be turned into a company. The project includes the design of a computer program (indicator) that facilitates market analysis for more efficient decision making. In addition it includes an explanation of the indicator coding process as well as the group methodology for trading which was tested under individual Demo accounts, providing promising results for a profitable trading strategy using the indicator created.

Table of Contents

Table of Figures

Chapter 1: Introduction

Taking a step back and observing the modern world, there is no question that trading is vital to the structure and survival of today's societies. Trading has been practiced by mankind for centuries and its implementation has evolved throughout years due to advancement in skill and technology. Initially, trading was done by exchange of good for goods, termed as barter trade that was done mostly between individuals and in a national scale. Gold, silver, and other "rare" metals began to be used as currency to facilitate trading and eventually paper money currency was introduced. Trading in currency became eminent to harmonize trade between countries. However different countries introduced their own currency which created complications in international trading because currencies differed in value and purchasing power.

To address such complications countries had to agree upon rules that would guide exchange rate. At first the Gold Standard was used which meant a country needed a certain amount of gold in order to print a certain amount of money. This however wasn't effective in the long run due to a relatively static money supply causing deflation. The Jason Hartman team explains this occurrence in their blog as follows: "Since the supply of money would only grow at the same rate as new gold reserves, it means that increases in economic productivity will cause deflation. The reason for this is because prices in an economy are the result of total real output relative to the total money in circulation."[1] (Jason Hartman Team)Deflation in a modern economy would be really bad because it could lead to what is called a "death spiral" where decreases in price would lead to lower wages, lower demand, lower production and eventually an economic recession.

The Bretton Woods system was formed in 1944 to set up a better system to fix exchange rates. In his paper "The Bretton Woods International Monitory System: A Historical Overview" Bordo, an economics professor and director of the Center for Monetary and Financial History at Rutgers University, stated that the architects of the Bretton Woods wanted a set of monetary arrangement that would combine the advantage of the classical gold standard such as exchange rate stability with the advantage of the floating rates while avoiding the defect of one or the other.[2] (Bordo) To make the system work, monetary institutions like the International Monetary Funds (IMF) were formed. Each

[1] (Jason Hartman Team)
[2] (Bordo)

country with different currencies had to set up an account at the IMF thus the modern Foreign exchange market (FOREX) was born. However, in 1971, the Bretton Woods System came to an end. This caused the dollar and other major currencies to become fiat currencies, bringing forth floating exchange rates. Fiat currencies get their value from government laws and regulations. The current FOREX market is based on floating rates which reflect the economic strengths and weakness of the country's currency. In 1994 currency trading started to be executed through the Internet, allowing almost anyone to be able to trade currency by using leverage. The focus of this Interactive Qualifying Project is how to succeed as an independent Forex trader by using MetaTrader 4 or TradeStation.

The complexity and innovation of modern societies has created the need for a structured way of trading. Such structure has been achieved through the various types of financial markets or exchanges which are defined by the specific asset class traded in them. An asset class is defined by Investopedia as "A group of securities that exhibit similar characteristics, behave similarly in the marketplace, and are subject to the same laws and regulations."[3] (Definition - Asset Classes) It is essential to have an understanding of various asset classes, the exchanges and markets in which they are traded before deciding what to trade.

1.1 Project Description

This project will give the basic guideline as to how one can get started with Forex trading. It will provide the reader with the necessary background information involving financial markets, their behavior and types of analysis that can be done. That information is then used to create a trading strategy utilizing an indicator created by the group that facilitates market analysis for more efficient decision making. The trading strategy is then tested by group members using demo accounts providing the reader with a description of how the general strategy can be modified to fit ones trading style and personality.

[3] (Definition - Asset Classes)

Chapter 2: Background Information

This section will cover the basic information needed to better understand the big picture of today's economy; including some of the various asset classes that can be traded, explaining the reasoning for choosing FOREX, and providing the basic background information concerning exchanges and trading platforms where these asset classes are traded.

2.1 Financial Markets and Asset Classes

A Financial Market is a very important component for the economic growth of countries around the world. It is a channel through which funds are transferred from one person to another, that is from lenders to borrowers thus, improving the economic efficiency. For example, an individual or more, who want to start a business but has no funds to start their business, can access those funds by borrowing from the one that has the funds and ready to lend it out. However, lenders and borrowers cannot meet in the air, they need intermediaries to connect them to each other and commercial banks play that role. Borrowers can access funds from lenders directly and indirectly through intermediaries that can be claimed in their future income or assets. As stated by Fabozzi J. Frank, when the buyer and seller interact in the financial market which provides a mechanism for investors to sell a financial instrument once price are determined for the traded assets, in other words, financial markets provide liquidity to the investor (Fabozzi 6). Fabozzi adds that the cost of transaction is reduced given the elimination of search cost and the information cost. So financial markets is what makes the trading of stocks, bonds, foreign exchange currencies and commodities possible since it is a platform that allows people to buy and sell the financial securities. This facilitates the smooth flow of income, allowing households, firms and government agencies to finance and invest their funds.

There is many types of financial markets: The Capital market which involves the trading of stocks and bonds, the Foreign Exchange markets which deal with the trading of foreign exchange currencies, the Money markets which deal with short term lending and borrowing of funds, and the Derivative markets in which Futures, Options etc. are traded. Robert O'Harrow Stocks describes a derivative as "a financial agreement between two entities that depends on something that occurs in the future, such as

the performance of an underlying asset. That underlying asset could be a stock, a bond, a currency or a commodity".[4] (O'Harrow)

The following subsection will give a more thorough description of the various types of financial markets. Since these types of financial markets are defined by the asset classes traded in them the second subsection includes a description of some major asset classes. The "Asset Classes" subsection doesn't include currency due to the coverage of this asset class in the Forex Market description as well as the rest of this paper. It is important to know that "international currency is the largest and most liquid asset class in the world"[5] (Currency: The Overlooked Asset Class)which is why this project focuses on FOREX trading.

2.1.1 Types of Financial Markets

FOREX

The FOREX market is the buying and selling of currencies and is one of the fastest growing markets in the world. From 2007 to 2012 the FOREX market has increased about 30%. It is the largest financial market in the world with an estimated $4 trillion in currencies traded daily. One of the reasons for this vast volume is that unlike other financial markets, FOREX is not tied to an actual stock exchange, it is an over the counter (OTC) market. Currencies are traded in pairs, with the United Stated dollar (USD), European euro (EUR) Japanese yen (JPY), British pound (GBP), and the Swiss franc (CHF) most commonly traded. For example, trading the value of USD against the value of EUR is considered the currency pair, "EUR/USD." The first currency located to the left of the slash mark is called the base currency, and the second currency that is located to the right of the slash mark is called the quote currency. FOREX being an OTC market it means that, when buying a currency pair, the base is bought and counter is sold. FOREX market transactions are performed either by dealers at banks or by FOREX brokerage companies. As the focus of this IQP, the FOREX market will be covered in more detail in a later section of this report but before diving too deep into the subject lets introduce some basic vocabulary that is commonly encountered when trading.

[4] (O'Harrow)

[5] (Currency: The Overlooked Asset Class)

Pip –Stands for price interest point and it is the smallest increment of price movement in a currency pair. Most currency pairs are measured to four decimal places. In the case of the USD/JPY, pip is measured to two decimal places. If the buying price of the USD is 1.6580 and then it shoots up to 1.6582 then you can say that the dollar has increased by 2 pips. 1 pip for the EUR/USD = 0.0001 and 1 pip for the USD/JPY = 0.01. When trading EUR/USD or GBP/USD, 1Pip = 10 US dollars.

In the forex market, the Spread is another component of trading. It is the difference between the Bid and the Ask price. For example if the USD/JPY read 2.3300/04, the spread would be the difference between 2.3300 and 2.3304.The Ask price is the price at which the trader buys the pair, it is also known as the offer price. The Bid price however is the price the trader sells the currency pair.

Leverage is a very key facet of forex trading; it is the ability to use traders' credit to trade on the forex market. For instance if a trader has $500 of margin in his account and he opens a$50,000 position, his or her leverage is 100:1, Augmenting either the traders profit or losses. How much leverage you are using will decide how much margin is required to open an account

Margin is the deposit required to open or maintain a position. The margin can be used or free. Used margin is that amount which is being used to maintain an open position and free margin is the amount available to open position. With a $500 margin balance in your account and a 1% verge condition to open a position, you can buy or sell a position worth up to an estimated $50,000. This allows a trader to leverage his account by up to 100 times or a control ratio of 100:1.

Capital Market

Capital market is design for the purposes of long-term transactions of securities by business, government and households. According to Valdone Darskuviene, Capital market is sector of the financial market where long-term financial instruments issues by corporations and government trade. (Ishaq Dar) Capital market has the original maturities of more than a year. It is the backbone of the country's economy. Without it institutions such as schools, factories, transportations (highways and railways) and homes would not be in existence. It is a market that individuals and institutions sell or buy securities (Stocks and Bonds) to allow firms or companies to raise funds for their institution.

Capital market involves equity and debt. The equity is where stocks of corporation are traded and the debt is contract outlining what and when the borrower should repay the lender. In this case the

agreement between the borrower and lender are made in order for the borrower to pay back the exact amount borrowed plus interest. Banks play a great role in capital market, lending funds to individuals, big firms and government institutions, thus providing resources for industries and business for long term needs.

Capital market is divided into; primary and secondary market**.** Primary Market facilitates the issuances of new securities. The primary market sole purpose is to raise financial capital support the new investment. For example if a corporation decides to sell a new issue, the corporation then receives the proceeds from the sales and this would be primary transaction. In contrast, secondary market facilitates the trading of the existing securities that allows change in the ownership of the securities.

Money Market

Money Market on the other hand is design for a short term lending and borrowing of funds mostly not more than a period of one year. Treasury bills, certificates of deposit, commercial paper etc. are the tools used to facilitate the borrowing and lending of funds. As stated in Investopedia the financial instruments traded in the Money Market are very liquid and extraordinarily safe, however most market securities trade in very high denominations this limiting access for the individual investor. [6] (Investopedia US)

Derivative Market

Derivative market deals with values derived from the values of an underlying asset. Robert O'Harrow Stocks describes a derivative as "a financial agreement between two entities that depends on something that occurs in the future, such as the performance of an underlying asset. That underlying asset could be a stock, a bond, a currency or a commodity". (O'Harrow)

The main traded instruments for derivative markets are Forward, Futures, Options, and Swaps.

[6] (Investopedia US)

Forward is a contract which two parties commit for a specified transaction and set date in the future. Forward contract obligates one party to buy the underlying at a fixed price at a certain future date from a counter party who is obligated to sell the underlying at that fixed price. (Rene). From the definition above, forward contract are contacts made for the future. Agricultural goods are the best illustration for forward contract. Buyer and seller contract can set up a price and time for the sale of goods that may be harvested in three month. The seller cannot change the prices, even if the prices of that good rose at the time they were sold to the buyer. It is an over the counter contract that means the contract may be negotiated directly between the buyer and seller.

Future Contract is more less the same as forward contract. It also involves an agreement to buy a fixed amount of goods at a specified price and time in the future. However, Future contract are standardized, that means future contracts are traded in a formal market place. The future contracts also require cash flow in case of a change in price to maintain the margin. This has been termed as mark to market.

Future and forward contract involves the right and obligation, but in Option contract, the buyer or seller has the right to buy or sell the good in the agreement, but has a choice whether to buy or sell the product at a specified price and time.

Swap contract is a contract between people to exchange cash flows in a future set date. Swap contract have two main components: the interest rate and cash component. Interest rate happens when two people agree to exchange interest rate periodically while currencies are the agreement to deliver one currency over the other currency. Swap currency involves an agreement between two people to trade in different currencies.

2.1.2 Asset Classes

Stocks

A stock represents a unit/share of ownership in a company. Companies sell stocks in order to raise money for further expansion. Selling stock is a more efficient way for a company to raise investment capital then taking a loan from a bank or by issuing bonds because it does not require paying back money or interest. In return the shareholder is entitled to an amount of assets and earnings of the company that is directly proportional to the number of shares he/she owns. This “ownership” status however comes with a lot of risk to the shareholder because they could lose a lot of their money if the company is not successful or all if the company goes bankrupt.

Stocks are traded in stock exchanges. Stock exchanges help facilitate the execution of trades between buyers and sellers. In order for companies to be listed on stock exchanges they must have a certain amount of earnings for a certain period of time. Once listed the company has huge opportunities to raise investment capital due to the infinite number of investors in today’s market. The New York stock exchange and NASDAQ are the biggest stock exchanges in the United States at the time. The New York stock exchange is particularly special because it is has human specialists doing trades while NASDAQ is known as a tech market where most of the world’s leading technology companies are listed on. On a later section we will be talking about the different platforms which are technically the tools used by traders to trade these exchanges.

Bonds

Another major asset class is bonds. Unlike stocks bonds are not equity. Bonds are technically a loan investors give to companies. Companies issue bonds to investors in order to raise investment capital and thus have to pay interest payment. In the trading world, this interest payments paid to the bond investors is called a coupon. Bonds also have a maturity date which is the date that the borrower (Company/Gov.) has to pay back the investor. The payment will include the original money borrowed as well as the interest rate accumulated over time.

A bondholder can however sell a bond before its maturity date. In order to make money by selling a bond before maturity it is crucial that interest rates of new bonds being sold to investors have lower interest rates. Having a coupon of higher value than other bonds in the market will increase other buyers' interest in the bond as well as its price thus giving the current bondholder the opportunity to sell his bond at a higher price than bought. Besides that, the bondholder can make money during the time they hold the bond because the return of a bond is in the form of fixed periodic payments that include interest (coupon). Being that the value of these fixed periodic payments is known in advance makes bonds' fixed income securities, thus generally safer than stocks. Also in case the company was to do badly in the market or go bankrupt bondholders are of higher priority when it comes to getting paid back compared to stockholders. However this lowered level of risk brings less reward since the bonds coupon doesn't increase if the company is very successful.

Even though generally speaking bonds are safer than stocks, one must be careful in their investments because that doesn't apply to all bonds. There are some extremely risky bonds in the market. These bonds are issued by unstable companies and lure buyers because of their high coupon values so it is important to take in consideration the issuer's stability in order to secure repayment. Bonds are traded through a broker because there are many different types of bonds with different qualities and maturity yields. Another reason why bonds aren't traded in exchanges like stock is because their process is affected by changing interest rates and credit ratings which makes it difficult to list a bonds current price.[7] (Morah)

[7] (Morah)

Futures and Options

It is not recommended that a trading newbie trade the following asset classes due to their complexity it is important to have an idea of what they. Having some knowledge about these asset classes would come in handy if one were to choose to diversify their portfolio.

An option as defined by Investopedia is “a financial derivative that represents a contract sold by one party (option writer) to another party (option holder). The contract offers the buyer the right, but not the obligation, to buy (call) or sell (put) a security or other financial asset at an agreed-upon price (the strike price) during a certain period of time or on a specific date (exercise date)”.[8] (Definition - Option). Just like with stocks a buyer would want the price of an option to go up if they placed a call order and they would want the price of an option to go down if they placed a put order.

When it comes to futures on the other hand the buyer/seller is obligated to buy or sell the financial asset at the agreed-upon price at the end of the contract. “Futures can be used either to hedge or to speculate on the price movement of the underlying asset. For example, a producer of corn could use futures to lock in a certain price and reduce risk (hedge). On the other hand, anybody could speculate on the price movement of corn by going long or short using futures”[9] (Definition - Futures)

Besides Bonds which are traded through a broker the other asset classes mentioned in this section are traded through exchanges and in today’s technologically advanced world any large corporation or individual trader could trade these asset classes in trading platforms such as Tradestation and Metatrader .

[8] (Definition - Option)
[9] (Definition - Futures)

2.2 Market Behavior and Trading Strategies/Systems

The behavior of the market can be trending, volatile or directionless. As Charlie Wright explains in his book "Trading as a Business", a trending market is characterized by a large, sustained increase or decrease in price. So if the market is making higher highs and higher lows it is said to be trending up and if the market is making lower lows and lower highs it is said to be trending down. The following graphs from StockCharts.com picture trending market behavior. (Radzicki)

Figure 1: Trending Market

A volatile market (Fig 2) is characterized by sharp jumps in price and a directionless market is characterized by small, insignificant up/down movements in price with a general sideways movement. When directionless, the market is referred to as being a sideways market. [10](Wright)

Fig.3- Sideways Market (fxs)

Fig.2- Volatile Market (Krivo, Quieting Market Noise)

[10] (Wright) the chart is under this footnote too

To achieve more successful trading it is best to have a system of trading systems that one can be able to trade in various market conditions. The chart below summarizes as to what type of trading system one should use based on market behavior and as we can see the two are highly dependent on one another. (Wright)

Market Type	Trading System
Trending	Trend Following
Volatile	Volatility Expansion
Directionless	Support & Resistance

Figure 4: Market Type- Trading System

Trend Following systems require the trader to trend trade which is defined by Investopedia as "trading strategy that attempts to capture gains through the analysis of an asset's momentum in a particular direction. The trend trader enters into a long position when a stock is trending upward (successively higher highs). Conversely, a short position is taken when the stock is in a down trend (successively lower highs)." (Definition - Trend Trading). Volatility Expansion as noted by the chart do well in volatile markets. They "measure recent volatility and attempt to buy an abrupt upside breakout with increased volatility or sell an abrupt downside breakdown with increased volatility." (Radzicki) Support and Resistance Systems are technically the opposite of Trend following systems because they are used when the market is Directionless.

It's a good idea to continue the jurney of creating a trading system by learning about the aspect of each type of trading system so one can make sure the time money and effort is put in a system suitable to the user. The chart below neatly summarizes aspects of each type of system to take in consideration. (Wright)

Table 1	Trend	S/R	Volatility
Time in the Market	Always in the market	Not always in the market	A substantial amount of time out of the market
Winning trades	Small percentage of winning trades	Higher percentage of winning trades	High percentage of winning trades
Where is money made	Money is made on big moves	Money is made in sideways markets	Money is made in market explosions
Where is money not made	Money is lost in choppy periods	Money is lost in trending periods	Money is not made in quiet markets
Biggest con	Many false signals, long drawdown periods	Difficult to sustain profit over the long term	Never get the big move
Biggest pro	Possibility of high profits	Higher percentage of profitable trades	High percentage of profitable trades
Profit	Average profit per trade high over long term, unlimited	Limited average profit per trade	Small profit per trade, limited
Philosophy	Buy high and exit higher, sell low and exit lower	Buy low and sell high	Very quick and short term trades
Emotional	Long sustained drawdown periods can be difficult	Easier to trade because you are buying low and selling high	Exciting to trade – trades are short-term
Type of Indicators used	Moving Average, ADX, price bands and channels	RSI, %R, Stochastics, Support/Resistance lines	Purely based on price

Figure 5: Aspects of a Trading System

The first thing on the above chart and one of the most important aspect to consider is the time a trader will be spend in the market. This aspect isn't solely focused on the amount of time in the market. It also has to do with the time of the day in which the trader will be in the market and the timeframe he/she will trade. In order to analyze and decide on which time frame to trade one must first decide what asset class to trade and much more. Any trading system, besides having a particular instrument that is being traded and a particular time frame over which trades are must have must have an objective, Entry and Exit Rules, Risk Management Rules, System Monitoring Techniques and Asset Allocation Rules. [11] (Radzicki)

11

In order to decide which instrument to trade one must take a look at all the different asset classes and decide which best interests them. For this project the group decided to focus on Forex trading not only because the forex market is one of the largest and most liquid markets in the world but also due to personal preferences. When taking part in forex trading one gets a knowledge of what is going on all over the world since the strength and behavior of currencies reflects the specific countries political and economic states. Fundamental analysis is used to study such aspect of forex and use it when trading. A thorough coverage of Fundamental analysis is shown in the next section, followed by coverage of another type of analysis, called Technical Analysis. Whether trader chooses to use Fundamental, Technical or both types of analysis in their trading systems the components mentioned in the previous paragraph are a must.

A trading system can have a single or multiple objectives such as: high annual return, high winning percentages, low time commitment, being in the market a small percentage of the time, not holding trades overnight or whatever other objective is more appealing to the individual trader. (Radzicki). In the methodology section of this project is shown a general strategy and the Trading System of different group members since it is important that a trading system fits the needs and personality of the trader. When it comes to trading systems, they need to be personalized to the individual that is using it. One size doesn't fit all, which is one of the main reasons why becoming a successful trader isn't easy, and it requires work and practice. A trading system must fit the trader's psychological profile and it takes time and practice to figure out the attitude one has towards risk and measure the ability one has to follow the rules of the system. As mentioned by professor Radzicki, "The rules of a scientifically developed, positive expectancy, trading system must be followed exactly, without exception. When this occurs, losses are cut short and profits are left to run." (Radzicki)

In order to help new traders get an idea of what how to come up with rules for a trading system the following subsections will explain what Entry and Exits are, Risk and Money Management etc… Given that this group focused on the Forex market, the Trading Sessions subsection focuses on the Forex market and includes characteristics and trading suggestions for each session. The System Overview section has some more information and suggestions that could come handy to new forex traders.

2.2.1 Set Up, Entry and Exit

Set-Up is the condition or set of conditions one must define before considering to enter or exit the market. These conditions are dependent on the type of trading strategy one choses to use and could be implemented through technical indicators. For example, trend following strategies would have set up conditions that involve the crossing of two different moving average indicators while support and resistance strategies could involve a price reaching the upper or lower lines of a moving average envelope.[12] (Wright) To come up with such conditions takes lots of time and practice as well as time spent researching and understanding different technical indicators. These set up conditions also need to be compatible with the timeframe in which the trader will trade so for example, for long term trading a trend following strategy would benefit from slower moving average indicators that analyze daily and weekly timeframes. By focusing on slower moving averages the system will be analyzing a broader view of what is happening and not focusing so much on the 5 or 15 minute price movements. After coming up with the set up conditions the trader is able to enter and exit the market efficiently. There is designated set up conditions for entry as well as for exits and they are usually inverses of each other. So for example when using two moving averages, the faster moving average crossing above the slower moving average indicates and entry, inversely, the faster moving average crossing below the slower moving average indicates an exit. One very important thing to keep in mind is that before entering or exiting the market one must confirm that the current price matches up with the direction indicated by the set up rule. Exits are important because they ultimately determine the size of one's profits or losses, the length of trades and amount of risk and ones total return. Exits tend to be more difficult than entries because when entering a trader enters on their own terms but must exit on terms set by the market, which is why it is necessary to have realistic expectations and take control of exits. (Radzicki)

[12] (Wright)

2.2.3 Risk Management

Risk management is a very important aspect of any and all trading systems. The amount of risk taken is directly proportional to both profit as well as losses in trading. Getting into risky positions could bring forth lots of profit if the market behaves as one predicted but it could also cause big losses if the market behaves otherwise. One of the best ways to manage risk is by setting Stop-Loss orders. As defined by Investopedia a Stop-Loss order is an order placed to sell a security when it reaches a certain price. Such an order is designed to limit an investor's loss on a security position. [13] (Definition - Stop Loss)

Stop Loss orders should be set between 25 and 30 percent below or above the entry price of a buy or sell order respectively. A stop loss order shouldn't be set too close to the price one paid because small fluctuations in the market would cause the order to get filled even though the market actually moves back in the direction the trader expected it to move causing loss in money or preventing the opportunity of more profit. There is a more complex stop loss order type called trailing stop loss order where the stop loss price is set at a percentage and follows the price of the market thus giving the trader the opportunity to take in more profit then if the stop loss was set at a fixed amount. Stop loss orders also shouldn't be further then 30% because that opens up room to a lot of risk. If the trade placed turns out to be a losing trade 30% of the money invested in that trade would be lost and it would be hard to make that amount back given that one has less money to make profit with. Losing capital is really bad because once ones total capital is smaller it would take higher winning percentages to make the same amount of money with less capital then it would take with a larger capital. So for example if one initially had $1000 and made a 10% profit it would mean $100 profit bringing the total capital to $1100. Now let's imagine this 10% was lost and the total capital is $900 after the loss, to bring the capital up to $1100 one not has to make 22.2% profit off the $900 capital they have.($\frac{x}{100} * 900 = 1100 - 900, x = .222$). In order to avoid large losses in capital one must position size by setting a limit to the size of each trade and amount that can be invested in one day base on the total capital.

[13] (Definition - Stop Loss)

2.2.4 Money Management

Money management is the process of budgeting, saving, investing and spending or otherwise in overseeing the cash usage of an individual or group[14] (Definition - Money Management) and is one of the most critical elements of forex trading. Money management if well planned can be a safeguard; it protects and enables a trader to stay in business.

Proper money managements requires a trader to develop a money management plan that involves limiting the maximum allowed lose for each trade. That means a trader should only risk a tolerable amount of their investment. As a trader there are fundamental money management rules that one needs to follow in order to be successful in their business. One should not be too greedy, anticipating that taking too much risk or overleveraging would result in higher return. That is a recipe for disaster. As stated in the article "*The Art of Surviving through Money Management*", "A market is a game of balancing fear, greed and hope. When a trader is out of balance, he or she is likely to lose money and if they are out of control, they will lose balance."[15] (Bressert) Another rule is that the trader should always let the profit run and cut loses short. They can that gradual taking out some money and it aside as they win. This protects the traders account from being wiped out in case the market turns against the traders.

While trading, one should be able to make objective decisions concerning the positions they open when trading. Avoid taking too much risk, but at the same time not allowing fear take. Money management is a vital part of a trading strategy thus It should not be overlook. When money management strategy is carefully followed, it can lead trader success in the business

[14] (Definition - Money Management)
[15] (Bressert)

2.2.5 Forex Trading Sessions

The forex market is a 24-hour trading market and it can be broken up into three major trading sessions. The European trading session which starts at the London open at 3:00am until noon EST, the North American trading session which starts at the New York open at 8:00am until 5:00pm EST and the Asian trading session which is considered to be from 5:00pm until 4:00am EST. The above trading session times are what one should pay most attention to however the chart below shows a more detailed description of the various treading sessions of the forex market (Note: There is slight changes due to seasonal time changes).

(Forex Market Hours)

Trading sessions according to EST (Eastern Standard Time):

Region	City	Open (EST)	Close (EST)
Europe	London	3:00 am	12:00 noon
	Frankfurt	2:00 am	11:00 am
America	New York	8:00 am	5:00 pm
	Chicago	9:00 am	6:00 pm
Asia	Tokyo	7:00 pm	4:00 am
	Hong Kong	8:00 pm	5:00 am
Pacific	Sydney	5:00 pm	2:00 am
	Wellington	5:00 pm	1:00 am

Figure 6: Trading Sessions (EST)

When developing a trading system it is important to figure out a particular time frame over which trades will be made since that greatly affects ones rate of success. When deciding the time of day in which to trade in it is important to first look at how the market generally behaves at certain times, what currency pair one is trading as well as what kind of a system one is using to trade.

Usually there is a lot more volatility and liquidity in times when various trading sessions overlap. So for example between 8am and 12pm is when the New York and London sessions overlap which is a very high volatility and liquidity time in the forex market. The chart below show the average pip movement of the major currency pairs during each trading session and it is noticeable that the New York and London sessions have high pip movements thus the overlapping of those two sessions would bring

forth large fluctuations in price. It is during this time period that important news reports from U.S and Canada are released which also contributes to the big moves.

Pair	Tokyo	London	New York
EUR/USD	76	**114**	92
GBP/USD	92	**127**	99
USD/JPY	51	**66**	59
AUD/USD	77	**83**	81
NZD/USD	62	**72**	70
USD/CAD	57	**96**	96
USD/CHF	67	**102**	83
EUR/JPY	102	**129**	107
GBP/JPY	118	**151**	132
AUD/JPY	98	**107**	103
EUR/GBP	78	**61**	47
EUR/CHF	79	**109**	84

(Trading Sessions)

Figure 7: Pair Pip Change per Trading Session

As mentioned earlier, the overlapping of these two sessions also creates high liquidity in the forex market. There is a high volume of trades at this time due fact that New York and London are the two largest financial centers in the world and the USD and EUR are the most traded currencies in the forex market. The EUR/USD, GBP/USD and USD/CHF represent the major currency pairs that involve the US and European countries thus market for these pairs would probably be more active at this time.

Trading in a liquid market facilitates easier entry and exit of positions which is one of the greatest advantages of trading at this time. One however should be careful of the volatility aspect of this session. Long term traders would probably want to avoid volatile markets the most since the large fluctuations in price might for example lead to a different entry position then the targeted one. Volatility is crucial to making profit for short term traders that don't want to hold open positions overnight but even they have to be careful nearing the end of this session since the market might get too choppy as European traders might close their position and lunch time approaches the U.S.[16] (Session Overlaps)

[16] (Session Overlaps)

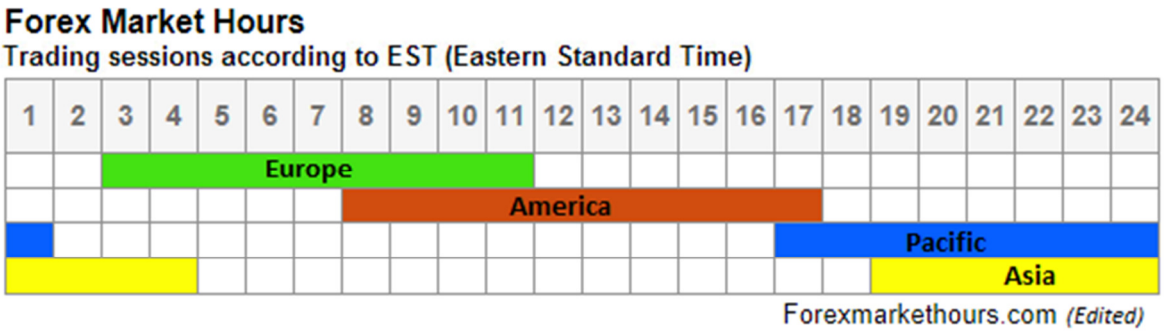


Figure 8: Forex Market Hours

As Figure 8 indicates, besides the Europe and American sessions, the Pacific and Asian sessions make up another large overlap. From 7pm to 2am is when the Australian and New Zealand markets of Wellington and Sidney overlap with the Asian markets of Tokyo, Singapore and Hong Kong. Within this timeframe liquidity is highest between 9:00PM and Midnight since Australia, New Zealand, Tokyo, Singapore and Hong Kong are all open. The AUD/USD, AUD/JPY, EUR/AUD, NZD/USD, AUD/NZD and NZD/JPY currency pairs often have active trading during this time. (forextraders.com)

When the Asian session is coming to an end the London session opens. London is the largest forex market in the world and it makes up 37% of the daily global forex trading. This is important knowledge because forex trends often start in the London due to the large percentage of volume associated with the London session. Distinction of trends is vital to technical trading thus it would be a good idea to trade and pay attention when the London session gets going. Even though this is of most importance to trend following type of systems one must not forget that going against a trend is a bad idea for any system. Going against trends poses greater risk as it basically means the trader is playing a guessing game instead of using a scientific method to make trading decisions. To make sure that a scientific way of making decisions is followed one must set up rules and conditions that have to be met before entering or exiting the market.

2.2.6 System Overview

Is high winning percentage a good thing?

An argument that is almost philosophical in nature is the debate of whether an individual wants a high winning percentage ranging in the 90% (0.90 range) which to inexperienced eyes is very appealing, or a winning percentage in the 70% (0.70 range) which might seem less interesting. A winning percentage does not necessary have a lot of value, apart from telling us a few characteristics of a system.

A high figure ranging in the 90's can directly express a systems volatility, meaning that the wins are constant and the method is efficient, but one mistake can catapult your system into a huge loss, such the case was with an investor we did research on who held multiple accounts for clients, this investor had a high winning percentage and was generating a lot of profit, but one day the market contradicted his methods and prediction causing him an 800 pip loss and had a few of his clients' accounts (Those who used higher leverage) wiped out.

A number ranging in the 70-80 range can show a less volatility in terms of the losses experienced, now that is less prone to being affected by a drastic change in the market.

A winning percentage does not hold much value in the sense that an individual can accomplish to have a 90% winning trade where accordingly only 10% would be losing trades, but what if the 10% of losing trades happened to wipe out your gains, this statistic can very deceptive. It can be of more value to look at the profit made from each individual trade and create a ratio of profit vs. loss in which (Profits/Losses) yields a profit factor.

Annual return

Many starting traders look at a 50% return per year, but most of them will not end up achieving this in the long term. Statistics have shown that most successful traders have annual returns near the 20% mark. At one point in his life Warren Buffet had become the second richest man in the world by accomplishing around a 25% percent returns compounded. This shows a steady system with less risks, but then again this is the reason why traders need to find a system that works for them and their risk and tolerance.

The systems behavior across different currency pairs

In an Ideal world, it would be perfect if a system worked across multiple currency pairs generating profit 24 hours a day. A system which analyses different currency pairs will be spewing out a lot of information for the trader, so unless the system is automated, the trader can choose to focus on only a couple of pairs to make his time more manageable and maintain an organized environment in which he can concentrate and preform.

FOREX trading and the behavior of any currency pair is directly proportional to the economic and political conditions of the countries and their respective monetary system. To trade any pair, it is beneficial to include research that involves technical analysis, international and national political relations, economic situations and the release of its data (Unemployment, GDP, Retail sales, etc.) and visual analysis (experience).

Time Commitment

Depending on a trader's personal experience and knowledge, it can take many sessions and several months to get a pretty general understanding of trading successfully. Trading takes time once you get started, and is strictly related to the individuals system. It all depends on how many times the trader enters and leaves the market over a certain time period. As a beginner it is important to ask: How much time am I willing to spend trading a day? Is it a hobby or a full time job? These are the questions that need to be answered by the individual him/herself.

It is possible to place trades in a part time basis and even place a few trades on the weekends, perhaps a trader can choose a currency pair whose session better fits his or her own time schedule. But most importantly for any starter it is crucial to give plenty of time to learn the market before jumping in. The goal for any trader would be to only place a few strategic trades throughout the month or the year that generate enough return for him/her-self or investors. Not only will the trader make money, but he will have plenty of time to live his life and dedicate time to other interests, or just his main job/source of income.

Holding Overnight Trades

As the name suggests, trading overnight involves traders holding their positions past the period for which their according session has ended, since the foreign exchange market never closes individuals are able to place any order at any time of day.

There are a couple factors that need to be pointed out since there can be both benefits and risks. All open and longs position being held after 5:00pm EST in the US are considered overnight. It is after this moment that the account belonging to the trader either received or pays out interest. An example provided by Investopedia is the following: 'If the Canadian interest rate is at 3.00% and the U.S interest rate is at 2.50%, the trader will then receive a payment that equals 0.50% into his account".

The 5:00pm time frame is always being held, even if a trader going in the market a minute to five, and exits the market a minute past five, the position will be considered overnight. But also if a position is opened after 5:00 pm and is closed at 4:59 of the next day, it will not be considered overnight.

Apart from this, Overnight trades involve risk for forex traders in the sense that the market may not necessarily operate under good conditions due to its session ending. But it poses a higher risk for stock trading, now that on X given day, a company can give a bad earnings report, announce the retirement of their CEO and announce financial issues now that the stock market has an open and a close.

2.3 Fundamental Analysis

Fundamental analysis is the study of the news and events that affects the performance of a currency of a particular country. These include political, economic and environmental information that has the tendency to affect price action. Traders use this information to determine the prospect of changes in price of currency pair.[17] (Forex Fundamental Analysis)

2.3.1 Economic Reports That Affect the U.S Dollar

Economic Data is the foundation of a forex trader. It is of extreme importance to maintain an economic calendar with easy access to the released information. Economic reports express directly the state of health of a country's economy

Trade Balance

Date of Release: Around the 19th of every month

Time of Release: International Trade – 8:30 am EST

Released By: Bureau of Economic Analysis

The United States trade balance report shows the difference in value between the exports of goods and services leaving the U.S. If the trade balance is positive, then the value of U.S exports is greater than the

[17] (Forex Fundamental Analysis)

value of imports coming into the country. And if the trade balance is negative, then it shows when import values are greater than export values.

A positive trade balance is proportional to an economy that is showing expansion, because higher export growth is a factor in positive trade reports and it paves the foundation for increasing expansion mainly in the industrial and manufacturing sectors.

On the opposite end, a negative balance can put pressure on a country's economy and currency.

Gross Domestic Product (GDP)

Date of Release: Last day of each quarter, and it is based on the previous quarter

Time of Release: 8:30 am EST

Released By: Bureau of Economic Analysis

The GDP is considered the broadest indicator of economic growth and output for the primary reason that it represents all the monetary value of services and goods brought forth by the economy. It takes inflation into account, thus allowing for a better comparison to previous time periods. It includes purchases made by the government, personal consumption, paid construction costs, the foreign trade balance and private inventories.

To traders in the market and other investors as well as analysts and policy makers, the GDP is directly proportional to the country's economic strength. A "normal" GDP growth that is considered healthy for the economy is around the 3% range, and on the contrary, two consecutive quarters of negative GDP mean the economy is going into a recession. Two great resources for foreign exchange traders are the sections on corporate profits and inventory.

Corporate Profits: A statistic that encompasses the net income and NIPA (National Income and Product accounts). The variables it used to come up with the statistic, include: After-Tax profits, Book profits and profits from current production.

Inventory: Items that are part of a business's assets that are ready to be distributed to customers or will be ready for distribution. This inventory is later turned into profit thus generating higher income for the company.

*These two are great in the sense that the indicate growth during the period, and breakdowns of all the major sectors of the economy.

Retail Sales

Date of Release: Around the 13th of the month

Time of Release: 8:30am Eastern Time

Released By: Census Bureau and U.S department of Commerce

Retail Sales are watched by investors now that this economic indicator gives the value in dollars of products sold to consumers. The release of this data covers the sales of the previous month thus making this a good indicator of money flow.

The Release of the Retail sales can cause volatility in the market. Its value is of great importance and can lead to investors to put or take out money of the market. Now that the retail sales are directly proportional to the growth of an economy. The slowdown of retail sales can signal a recession now that it means the population is spending less.

Employment Data

Date of Release: First of the Month

Time of Release: 8:30am Eastern Time

Released By: Bureau of Labor

Employment is one of the important indicators that how the health of the economy. The employment announcement is always done on the first Friday of every month at 8:30 am EST. This announcement includes the unemployment rate; which is the percentage of the work force that is unemployed, the number of new jobs created, the average hours worked per week, and average hourly earnings. This report often results in significant market movement. For example recent news that unemployment rate in the US has dropped to 7.7 percent may help strengthen the economy and the strength of a dollar as well.

Housing Data

Date of Release: Around the 17th of each month

Time of Release: 8:30am Eastern Time

Released By: U.S Census Bureau

Housing data another indicator use to determine the strength of the economy. The housing data includes the number of new homes that a country began building that month as well as existing home's sales. Residential construction activity is a major cause of economic stimulus for a country and so it's widely followed by Forex participants.

2.3.2 Inflation

The inflation of a currency measures the change in the price of goods over a certain period of time, most commonly a yearly basis. An increase in the inflation number can be noticed if the prize of a gallon of milk for example went up compared to its price from the previous year.

Inflation can be caused by an oversupply of currency, and it is also a sign that the country is importing more goods than the ones being exported. The Fed then turns to lowering the discount rates, which lowers the rate at which the banks are borrowing dollars, which in turn puts more money into the hands of the public and the economy; hence these same banks lend it out at higher interest rates. When more dollars are printed or when more keys are stroked for "digital" currency, the value of dollar decreases, since there would be more of these flowing through the system

The higher the Inflation rate, the less one will be able to buy in comparison with other currencies. On the other hand, a country with a consistent low inflation rate shows a rising value for its currency now that its purchasing power also goes up. And not forgetting to mention, a lower inflation keeps prizes at a lower level for consumers.

Several Economic reports are used to measure inflation, One of these including the Consumer Price Index and the Producer Price Index which tell the cost of goods in comparison to previous figures, Meaning that if the CPI and PPI shows an increase in prizes, one of the main reasons can be the inflation, which can be calculated by looking at a comparison of prices.

As time goes on both the Producer Price and Consumer Price indexes will show a similar percentage in their rate of inflation. But it is also important to note the CPIs follow the PPI in terms of increasing or decreasing, and this is the reason why investors choose to mainly follow the CPI.

Many countries that are highly dependent on the exportation of one particular product, or one particular sector, the scarcity of this product which can be affected by a plague, or natural disaster, can have a huge effect on the figures produced by the producers, and the amount spent by the consumers.

2.3.3 Interest Rates

Interest rates are set by banks and is the most important in the field of forex trading, it is the main drive in the Forex market; all of the above mentioned economic indicators are closely watched by the Federal Open Market Committee in order to gauge the overall health of the economy. The Federal

government can use the tools at its disposable to manipulate the currencies. It lower, raise, or leave interest rates unchanged, depending on the evidence it has gathered on the health of the economy.

2.3.4 Political Conditions

The political state of a country, including their economic policies and plans or relations with one another plays a huge role in the value of a currency. It is very important to maintain good track of the political news and future events. Monetary policies are extremely important factors in economic decision making, now that these cause changes in the interest rates. Forex Markets if guided by economic reports and political situations which indicate a country's economic strength. Knowing the changing factors and how to react accordingly is extremely important for any trader.

Stability

In every Investors plan, there has to be a place to park money when the market seems unstable and too volatile. In these situations many FOREX traders decide to place money in currencies that provide safety in times of uncertainty. Behind a currency that provides this type of safety, there is always a country whose power is affected by its political and fiscal conditions, their central bank policies, and their purchasing power. For example, when the crisis in Egypt occurred and political tension threatened with a meltdown in the Middle East, the US dollar rallied making it a safe haven. Despite many legitimate concerns about the US economic recovery and the effect of massive debt on the long term value of the dollar, the dollar, for many, was the best of choices, and a similar situation was most recently seen with the economic crisis in Cyprus, which put a lot of pressure and uncertainty on the EURO, making the USD a good currency for investors at the time.

Along with the U.S dollar, the Swiss Franc is considered a safe haven, Not only in the sense that it a stable currency, but also their banking system held a Zero percent 0% interest rate and required very low information from depositors, thus allowing for a strong economy. Customer confidentially provided by these banks, makes it difficult for foreign and local authorities to access information about fraudulent account holders. Just recently the Swiss banks introduced a negative interest policy as a result from pressure of the US government, to deter people who evade taxes. As an example, if the interest by the Switzerland bank was set at negative 3%, an individual who deposits 100,000$ into an account, will have 97,000$ by the end of the year.

2.3.5 Forex and News

Forex News trading is taking advantage of the anticipation reaction and follows through of major news events. Major news events are events which significantly alter what market we already know of. Events such as natural disasters and war. And more importantly so the release of events, each and every major country has something called economic indicators, these indicators reflect weather a country is doing well, and what sector is doing well. This information is released in an organized predetermined factor months in advanced.

Why bother with news trading?

There are certain news events that are priced into the market, but there is a surprise factor that causes a change and a reaction which could be used to tell you where the market is headed. Forex news trading is time based, if news are set to be released by a certain time, one can prepare for such in an also organized fashion, it gives quick results, and that coupled with Se times, it can give low risk and high reward setup, One risks for example 20-30 pips with a target of 2 or three times that. It also provide with a higher profit factor. Define Profit factor.

Types of News Trading

Pre-Event trading – Mainly sentiment based, pre-event trading operates on creating an anticipation that news will be wonderful/bad based on what the market sentiment is about. There are many companies who focus on market research in mainstream media, with the main purpose of predicting the possible outcome of the news. They give the information before hand and usually have an accuracy of about 80%, these provide no pressure entries, but on the other hand these require a high amount of knowledge and experience on handle.

Event-Trading – Many traders would rather not deal with the sentiment leading up to the release of the news, But rather choose to trade on the event itself, thus leading to a lower Profit risk factor, Meaning that the profits can be lower and the risk of losing money is lower also. This applies to the entire market, but it is significantly important in Foreign exchange now that it causes very distinct fluctuations.

Post-Event trading- This type of news trades focuses mainly on longer terms profits, assuming that a certain event will affect a currency pair positively/negatively throughout a longer period of time.

Analysis of Past Press Releases' effect on the Market

A few Examples of market behavior in reaction to news releases can be seen below.

Retail Sales

Retail sales figures for the United States in relation to the EUR/USD chart had a very distinguishable pattern that followed through 4 straight repetitions during the research for this project. By looking directly at the date of release in relationship with the behavior of the market the following was noticed: a higher than expected percentages favoring the US currency and smaller than average percentages lowering the price of the dollar.

History	SMQ	Actual	Forecast	Previous
Jan 15, 2013	0.6	0.5%	0.2%	0.4%
Dec 13, 2012	0.6	0.3%	0.5%	-0.3%
Nov 14, 2012	0.6	-0.3%	-0.2%	1.3%
Oct 15, 2012	0.6	1.1%	0.7%	1.2%

Figure 9: Retail Sales

(www.ForexFacroy.com/calendar.php)

Every time the figure was green (raised the value of the dollar), we see a negative candle stick in the EUR/USD and on the contrary, every time we see a red figure above (lowers the prize of the dollar), we can see a green bar in the EUR/USD chart.

Figure 10: Market Behavior to Retail Sales

European Forex Economic Calendar

The following analyses based on the European Forex Economic calendar shows how certain press releases affected the market.

7 February 2013 – Monetary Policy Decisions

On the 7th the ECB decided to keep interest rates the same. Decisions to keep interest rates unchanged suggested that the Euro economy was still struggling and the ECB would be trying to help out the economy by not changing interest rates. This news would affect the euro negatively since buyers of a currency are attracted to currencies of reliable economies.

At the very moment this news was released the euro was still going strong which was somewhat worrying for the weak economy and its recovery due to possibility of hurting exports. Later that day Mario Draghi said in a news conference that the door is open to stimulus measures if needed by Europe. Suggesting that if necessary Europe just like Japan would take part in monetary easing to bring the value of the euro down, however according to articles from the WSJ that would have been just a method he was using to talk down the euro which indeed worked.[18] (Blackstone)

The graph bellow of EUR/USD in the 30 min timeframe shows a drastic fall in the EUR caused by Mr. Draghi's speech and the ECB decision to keep interest rates unchanged.

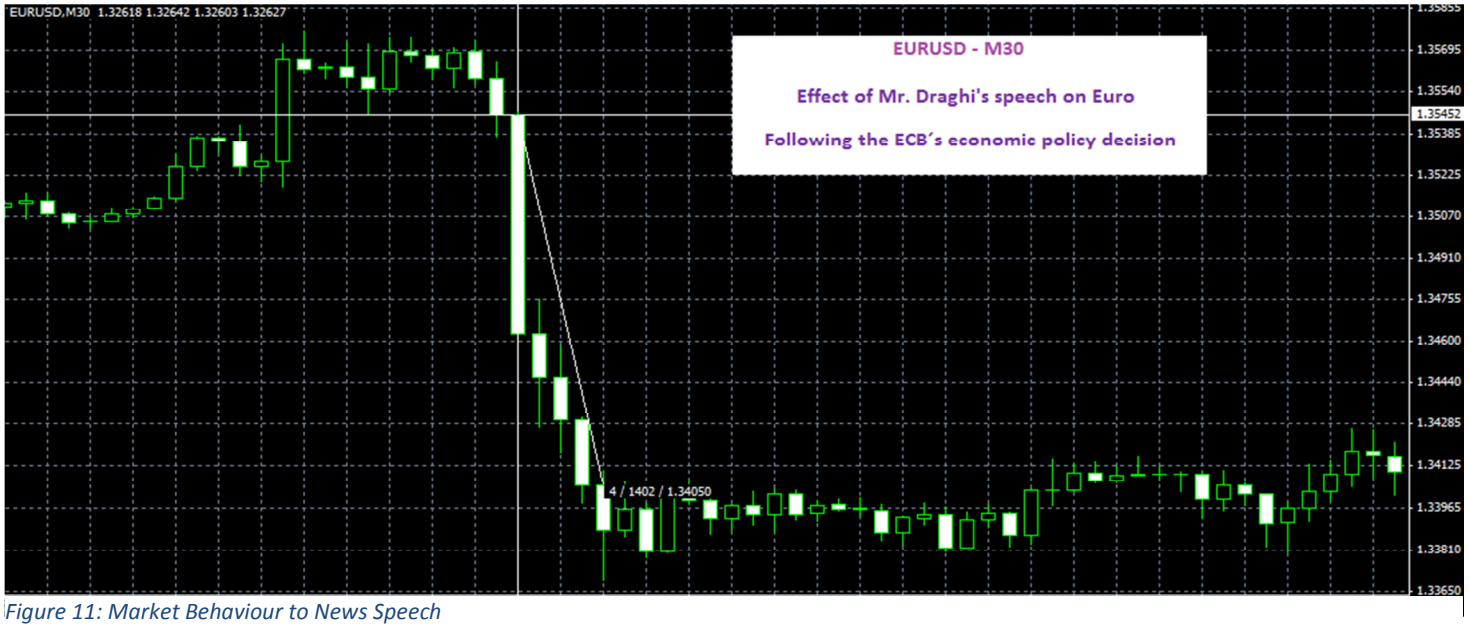


Figure 11: Market Behaviour to News Speech

[18] (Blackstone)

Looking at the chart it is noticeable that if one were to closely follow the news there would be enough time to react to it and make profit after these announcements.

14 February 2013 – Germany GDP for 4th Quarter

Another important Economic Release that Affected the forex market was the seasonally and calendar adjusted German GDP for the 4th quarter of 2012. Germany is the main Drive in European Union economy, thus news relating to Germanys economy have great influence in the euro. It was predicted that there would be a contraction in GDP for the 4th quarter however the contraction was larger than predicted, causing a bearish market for the euro. The effects of this news are demonstrated in the chart below by the circled area on the right.[19] (Economic Calendar)

Figure 12: Market Behaviour to GDP

20 February 2013 EUR Consumer Confidence

[19] (Economic Calendar)

On February 20th the consumer confidence was released by the European commission which was lower than the predicted consumer confidence. [9] (Economic Calendar) This news would affect the Euro negatively and that is exactly what it did. The chart below shows how the Euro went down considerably. In the chart we can also see that if one were to follow the news in a timely manner there is enough time to react and make profit.

Figure 13: Market Behavior to Consumer Confidence

2.4 The Genesis of Technical Analysis "The Dow Theory"

Technical Analysis is a very essential tool in forex trading used to predict the price movement and projecting the future market trends by studying past history of the charts in the previous market action. Every trader resorts to technical analysis as an instrument for analyzing the market behavior to be able to know when to enter and exit the market. This type of analysis was developed in the context of the stock market however, since 1970; it has been adopted by forex traders to study the currency market. (Paul)

The foundation of technical analysis originates from the Dow Theory that evolved from a series of articles published in the Wall Street Editorial between 1900 and 1902 by Charles H. Dow. Charles Dow was a Journalist and editor of the Wall Street Journal and a co- founder of the Dow Jones industrial and transportations. (Langager and Murphy)

Dow basically held the notion that asset prices revealed the basic fundamentals and business conditions and by studying these dynamics, traders can ascertain the behavior of major market trends. Dow's Theory focused on the stock market trend as an indicator of complete market situation in the economy. The idea behind the theory was that an increase in business activities of the Dow industries leads to a spike in Transport activity. His claim was that industrial firms make products and the products are shipped by transportation companies so their operations are positively correlated. Dow Theory has been categorized into 6 basic principles. The first of tenet states that information is discounted by the prices. All pertinent information about the asset is integrated into its price history, so it is not necessary to forecast the fundamental factor that affects asset's value.

The second tenet is the price movement trend and is defined by Dow as a situation in which each successive rally closes higher than the previous rally high and each successive rally low closes higher than previous rally low.

Dow divides the trend in three parts: the primary trend, the secondary trend and Minor trend.

The Primary trend is the main force behind the trend. It has the upward and downward movements known as the bull and the bear markets. The movement can last for one or more years before it can return to its original state. The Secondary Trend mostly takes a shorter period from three weeks to three month. Unlike primary trend that moves in a straight direction, secondary trend moves

in opposite direction. And lastly the third trend is the Minor trend. The Minor trend is the shortest, it last less than three weeks.

The third of the principles mentions that the primary trend has three phases. These phases are the accumulative phase, the public participation phase and the distribution phase.

Accumulative Phase as described in the Investopedia, is the start of the upward trend, it comes at the end of the downtrend when there is pessimism. (Langager and Murphy Sec.4). At this stage the investors are comfortable buying stocks or currencies in the hope of selling it when the price rises.

Public participation phase happens when the investors have entered the market. The investors enter the market at the accumulative stage as they assume that the risk is minimal and the recovery foreseen. Distribution phase is the selling stage, that means the market is strong and the investors begins to sell off , that means they would be predicting the downward movement. A wise trader, who has been watching the price movement, would begin to sell of their assets, but for the ignorant traders, they would buy expecting the market to stay strong. However that is not always true as the market moves up and downward.

The Bear Market on other hand is the downward movement. It consists of distribution, panic and disappointment. This is when the prices go down and the trader is forced to sell their asset at the lower price than they bought it in the hope to gain a portion of their assets.

The Averages must confirm each other is important tenant. According to Charles Dow the growing industry index cannot be sustainable as long as the transport index did not rise. The idea behind this is that transportation as important as industrial firms. As stated at the (RguruKool website) that Factories had to ship their goods to market, mainly through rail. Even Dow's first stock averages were in index of industrial and rail companies. Dow believed that the movement from bulls to bears would not happen or signaled if both indexes are not in the same level. This follows the Dow Jones Industrial Average (DJIA and the Dow Jones Transportation Average (DJTA) theory. According Dow Theory averages reflects all the information, experience, knowledge, opinions and activities of all stock market investors.

The trend should be confirmed by volume. Dow Theory is dependent on the price movement to predict the change in the market. The relationship between the price and volume is very important. In this case if the high volume is characterized by the upward movement of the market that means that

price will move higher. The volume can be at the beginning, the middle and at the end of the trend. Volumes always help to determine whether the existing trend is healthy. As forex trader, it would be great help to watch the, volume, trend price movement; because this can determine the value of the currency they are trading on.

A trend is assumed to be In effect until it gives definite signal that it has reversed. This tenet of the Dow Theory is in direct correlation with Newton's law of motion which states that an object in motion tends to remain in motion unless external force is applied to it. So the market, especially the primary trend that moves in one direction and does not change unless interrupted by external forces. As stated by (Murphy John 1999) this forms much of the foundation of modern follow-approaches. According to Murphy, the tenet relates to physical law of market movement (Murphy 28)

As much as the Dow theory deals with trends, volumes, price movement, charts, moving average, a forex trader must remember that technical analysis rooted in the Dow Theory was developed for the stock market and so it emphases on the primary trend that that flows in a particular direction. Currency trading is volatile and so the trend could change at any time. Secondary trend and Minor could be of great importance to forex traders because they have to "pay great deal of attention to minor swings for the timing purposes" (Murphy 32). However, Dow theory is of technical analysis is the important tool for forex traders. In order to be successful in their trading business, traders need to pay close attention to trend, volume and price movement in order to buy or sell the currency.

2.5 Technical Analysis

Technical analysis is the study of market action using charts and statistical analysis to forecast future price movement. The price chart is the primary tool and based on the market action takes everything into account, which means that price reflects every factor that could possibly influence the price. It also identifies the trend at which the price is moving. Price can be identified as up, down or sideways as any point of any given time scale. It helps to understand the future by studying the past. For example if the certain price chart patterns has appeared regularly in the past and has usually been followed by a move in a particular direction that it gives the trader a level of confidence that the repeated patterns will lead to a similar consequent price move

2.5.1 Moving Average Indicators

Moving average indicators are among the simplest and easiest to use of all indicators. There are various types of moving average indicators that use different ways/formulas to essentially plot the average price of a security over a set amount of time. This technical analysis tool is used by traders to smooth out fluctuation in price of the security being analyzed.[20] (Definition - Moving Average)By doing that it becomes much easier to see trend or cycle of the security which is why moving average indicators are used in trending markets. The most popular moving average indicators are Simple Moving Average indicators (SMA), Exponential Moving Average indicators (EMA).

Simple Moving Average

[20] (Definition - Moving Average)

The Simple Moving Average indicator is used to smooth out the price volatility so a clearer trend is apparent. Its calculation is simply the summation of the closing prices over a period of time divided by the number of prices being summed up.[21] (ChartSchool - Moving Average)

$$SMA_c = \frac{\sum_i^c P_i}{n}$$

SMA_c = current simple moving average

$$i = c - n$$

n = period

The simple moving average indicator, as well as other moving average indicators, is very easy to use. They can be used by simply looking at the direction of the moving average to look at whether the security/currency is in an uptrend or downtrend. It can also be interpreted as a signal by looking at the position of the actual price compared to the moving average. If the price is located below the moving average it shows that the currency is in a downtrend therefore the crossing of the price from above to bellow the moving average would be a possible sell signal. When the price is located above the average it means that the currency is in an uptrend therefore the crossing of the price from bellow to above the moving average line would be interpreted as a buy signal. The SMI however it isn't very sensitive to quick changes in the market which is why it's better used for longer term periods. The Exponential Moving Average indicator tends to be used more by traders.

Exponential Moving Average (EMA)

Exponential Moving Average (EMA) indicator could be much more efficient than a SMA indicator. Unlike the SMA where all the data used is weighted equal, the EMA indicator gives more weight to the most recent data. Today's economy is constantly changing so it would be to a trader's interest to give more importance to the most recent data. By doing that the EMA indicator reduces lag. (ChartSchool) In order to initiate its calculations an EMA starts off by taking a SMA and the amount by

[21] (ChartSchool - Moving Average) – Moving Average, Simple and Exponential

which an EMA indicator weighs the most recent prices depends on the time period used. For longer time periods the EMA's weight factor is smaller than that for longer time period EMA indicators.

The weight factor for the EMA is calculated with the following formula:

$$W_f = \frac{2}{1+N}$$ N = Period

In order to calculate EMA one must use the following equation:

$$EMA_c = W_f(P_c - EMA_{c-1}) + EMA_{c-1}$$

EMA_c = Current Exponential Moving Average

P_c = C**urrent Price**

EMA_{c-1} = Previous Exponential Moving Average

This indicator is better fit for short term traders than the simple moving average because it responds faster to changes in trend due to its stronger representation of most recent data. However, because the EMA is more sensitive to changes, it could be prone to false signals. For that reason one must incorporate different indicators in a trading strategy in order to make it more reliable.

Moving Average Envelope indicator

One way to improve Simple and Exponential Moving average indicators incorporate them with a moving average envelope indicator. This indicator works just by adding and subtracting from the moving average a certain percentage of deviation. Two parallel bands are created below and above the moving average by this certain (user defined) percentage of deviation. (Upper Band = MA + % deviation, Lower Band = MA - % deviation)

Moving Average Envelope indicator is essentially used like the Bollinger Bands indicator. Its Bands show overbought and oversold levels, trend, and indicate price breakouts. The chart below is a great example of the use of Moving Average Envelope indicator. [22] (Moving Average Envelopes)

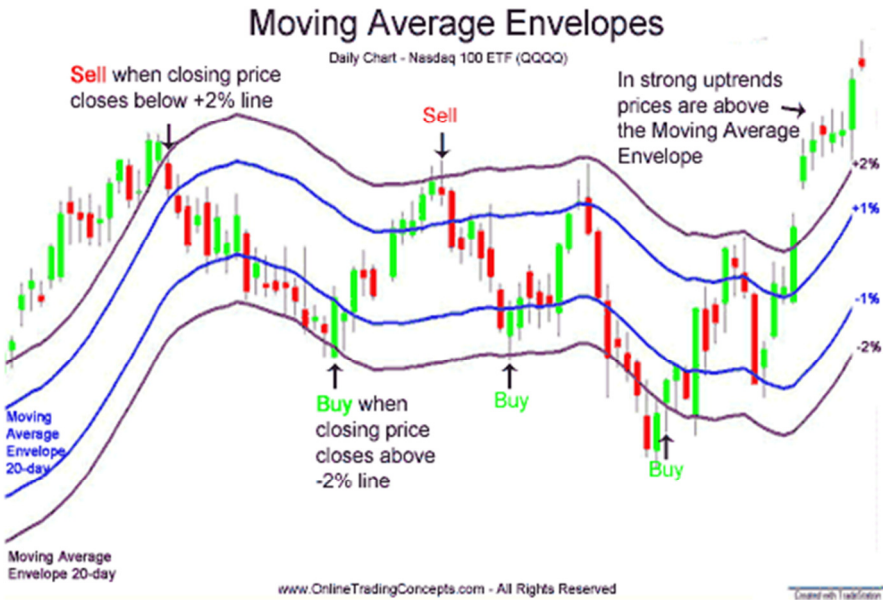


Figure 14: Moving Average Envelopes

What eventually determine the efficient use of this indicator are the parameters one uses. These parameters depend on the traders or investors objectives and characteristics of the security being traded. For longer term investments it is better to use longer period moving averages with relatively wide envelopes while traders are better off using shorter period moving averages with tighter envelopes. Also it is better to use narrower bands for securities with low volatility and wider bands for securities with high volatility because the securities with high volatility need bands that cover more price action.[23] (ChartSchool - MA Envelopes)

[22] (Moving Average Envelopes)
[23] (ChartSchool - MA Envelopes)

2.5.2 Alligator Indicator

Another moving average based indicator is the alligator indicator. It is made up of three different period based moving averages that form the jaw, teeth and lips of the alligator. The jaw of the alligator is created by the longest, 13 period candlestick moving average. The alligator's teeth line is created by an 8 period moving average and the alligators lips are created by a 5 day moving average. To create the jaw, teeth and lips of the alligator these moving averages are shifted forward (in the future) by eight, five and three bars respectably. The chart below shows an example of the alligator indicator used for the currency pair GBP/USD.[24] (Sabodin)

(Blue= jaw)

(Red= teeth)

(Green= lips)

Figure 15: Alligator Indicator

According to Bill Williams, the inventor of the alligator indicator, when the three moving average lines are intertwined they represent the alligator sleeping, so when the alligator sleeps for a long time he wakes up hungry. With that said, once the lines start untwining there is an expectation of high price movement in a certain direction or another. When the lines are in an uptrend (green above red, red above blue) it is interpreted as a buy/ long signal. And when the lines are on a downtrend (green below the red, red below the blue) would signal a sell/ short position. When the lines start getting closer together again it means that the trend is getting weak again and as they intertwine the alligator is falling asleep which means its best to stay out of the market.

[24] (Sabodin)

2.5.3 Aroon Indicator

The Aroon indicator is an oscillator type indicator that shows whether a security is trending or not and how strong it's trending. It more specifically shows when a new trend is emerging which is how it got its name which means "Dawn's Early Light". The Aroon indicator is made up of an Aroon-Up and an Aroon-Down indicator. The Aroon- Up indicator measures the number of periods since price recorded an x-day high while the Aroon-Down indicator measures the number of periods since price recorded and x-day low. The following equations describe mathematically the Aroon-Up and Aroon-Down indicators.

$$ArronUp = 100 * \frac{N - n_{maxN}}{N} \qquad ArronDown = 100 * \frac{N - n_{minN}}{N}$$

The Aroon indicators as it is noticeable by the equation are shown in percentage and can fluctuate between 0 and 100. [25] (ChartSchool - Aroon)

The Aroon indicator is quite simple to interpret to a certain extent. If the Aroon-Down is below 50 and the Aroon- Up above 50, it would be an indication of a bullish trend. If on the other hand the Aroon-Up is below 50 and the Aroon-Down above 50, it shows tendency for a new x-day low rather than high which indicates a bearish market, thus a short signal. To recognize the trend of the security/ currency pair one must focus on the 70 and 30 values. The Aroon-Up being above 70 and the Aroon-Down below 30 indicates a strong uptrend while the Aroon-Up being below 30 and the Aroon-Down above 70 indicates a strong downtrend. [26] (Aroon Indicator)

To recognize the emerging of a new trend using the Aroon indicator one must look for the following signals. They shall first see the Aroon lines cross each other. After crossing the Aroon lines will cross above 50 is a new uptrend is emerging or they will cross below 50 is a new downtrend is emerging. Lastly the emerging trend will be clearer when the Aroon lines reach 100.[25] The chart below shows the Aroon Indicator.[i27] (Definition - Aroon Indicator)

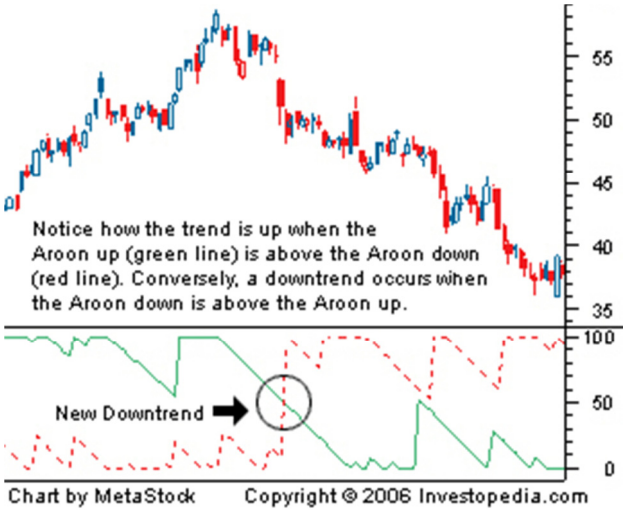


Figure 16: Aroon Indicator

[25] (ChartSchool - Aroon)
[26] (Aroon Indicator)
[27] (Definition - Aroon Indicator)

2.5.4 Relative Strength Index

The (RSI) or relative strength index is one of the most popular indicators used by the Foreign exchange traders. Initially made by J. Welles, its purpose was to measure the strength currency pair. This indicator is calculated by the formula[28] (Definition - RSI):

$$RSI = 100 - \frac{100}{1 + RS*}$$

Where Rs = Average of x days' up closes / Average of x days' down closes

It compares a currency pair's updated value against its past performance, or a currency pairs' high days against its low days. RSI operates on a scale that ranges from 1 to 100, where any point above 70 is considered overbought (Sign that the price of the asset is becoming overvalued), while any point below 30 is considered oversold (A Sign that the asset has dropped below where its true value resides) and it most commonly operates at a rate of 14 period. And it's most commonly used for the two following purposes:

- Its main purpose is to help recognize Reversals in the trend and rare conditions to watch out for. In practice, a Relative strength index over 70 would indicate a sell position, assuming that the value of the pair will drop and on the contrary, an RSI below 30 is seen as over-sold which would indicate a buy signal assuming that the prize will rise looking for stabilization.

- It also has the ability of being used to indicate divergence by locating the positive and negative values between the currency pair being used, and the RSI. In application, a raising currency pair for which the RSI drops from a high point down to low point rapidly. The currency pair will often reverse directions after such behavior. Below is a screen shot of the RSI indicator used in the platform "TradeStation" which we are currently using in our IQP.

[28] (Definition - RSI)

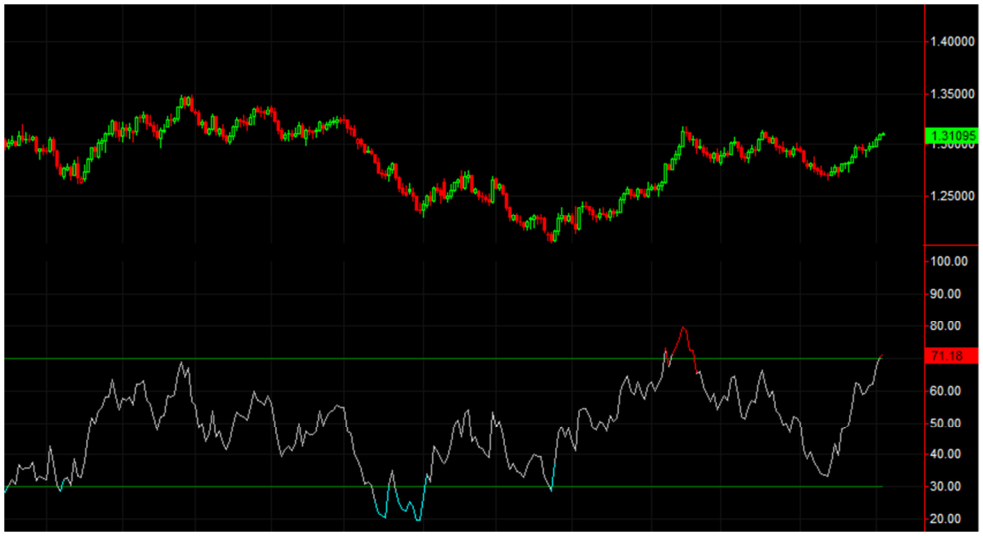


Figure 17: Relative Strength Index

This Figure shows the output of the RSI, the red color in the bottom graph shows values above 70, and the blue color in the bottom graph signals when the value drops below 30

2.5.5 Trix (Technical analysis)

Deriving from the name Triple Exponential, the TRIX works as technical analysis oscillator, by showing the slope of a triple exponential moving average. Similar to other moving averages, the TRIX smooths out data and price, therefore making it a trend-following indicator. Accordingly, a rising line is relative to an uptrend, and a falling line is relative to a down trend. What makes this indicator different than the others is the triple exponential, meaning that it applies to the results being spread much more broadly.

The TRIX formula goes as following[29] (Milton):

$$MA1 = MA1n-1 + [(2 / (n + 1) * (n - MA1n-1)]$$
$$MA2 = MA2n-1 + [(2 / (n + 1) * (MA1n - MA2n-1)]$$

$$MA3 = MA3n-1 + [(2 / (n + 1) * (MA2n - MA3n-1)]$$
$$T = (MA3n - MA3n-1) / MA3n-1$$

[29] (Milton)

The equation compares current and previous moving averages. This indicator combines both trend and momentum, making it a bit more volatile and better suited for center line crossovers. The main usage of its ***center line*** which is one of the most important parts of this indicators, apart from the oscillations themselves, it is seen as the centerline and can produce both good and bad signals. The idea with this center line is so visualize the previous momentum of the line in order to have an edge as to where it will head, for the sole purpose of maximizing profits and minimizing loses.

Apart from using the centerline and oscillations for trend following, it can also be used to indicate divergence, by bearish divergences signaling short entries, and bullish divergences signaling long entries

Figure 18: TRIX

Figure 18 shows the TRIX technical analysis tool being used in TradeStation, center line is represented by green

2.5.6 Price Channel

This helpful technical analysis indicator helps the trader identify sell and buy signals based on prices breaking out through a determined parameter. The channel is defined by two lines:

- **Upper Price Channel:** It is first identified by the user over which time period it is implemented, it determines the highest-high over this interval and plots them in the graph.

- **Lower Price channel:** Similar to the upper price channel with the only difference that it identifies the lowest-lows.

It is identified by the following formula: for a monthly period,

Upper band: 30-day high

Lower band: 30-day low

$$\text{Center Line} = \frac{30 \text{ day High} + 30 \text{ Day Low}}{2}$$

Price channel Formula ^ [30] (ChartSchool - Price Channel)

When a closing price falls above the upper band (in FOREX), it signals to sell, assuming that the price will fall, and accordingly, when a closing price falls below the lower band it signals to buy, assuming the price will rise, due to support from earlier values. As seen in the following picture.

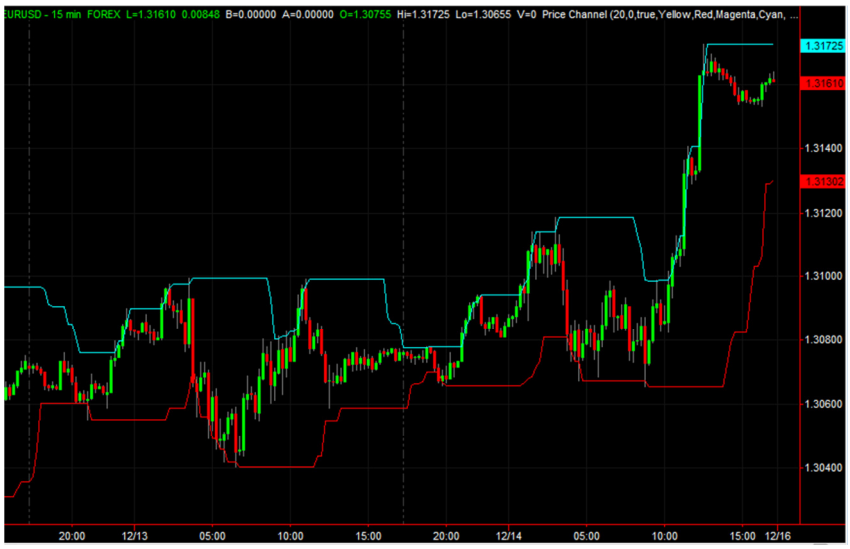


Figure 19: Price Channel

[30] (ChartSchool - Price Channel)

2.5.7 HRI

Also known as the Herrick Payoff Index, this indicator uses open interest, volume and price so signal divergences in the value of options contract or futures; the HPI is based on two concepts.

- **Rising Open interest:** At the time when interests and prices rise, it signals a bullish market that confirms a rising change in price. Similarly when these prices drop and interest goes up, it confirms a bearish market by the falling prices.

- **Falling Open Interest:** This is known when the interest drops and the value rises, this is signaled to be a bearish market which does not permit reversals. Likewise when the value along with the interest drops, making it a bullish sign with potential for reversal.

The HRI is represented by the following formula:

$$\frac{Ky + [(K' - Ky) * S]}{100000}$$

$$Ky = The\ previous\ period's HPI$$

$$K' = (C * V * (MJ - My))[1 + \frac{2 * I}{G}]$$

$$S = The\ multiplying\ factor$$

C = the "Value of a one cent move"

V = the day's Volume

I = the absolute value of the difference between todays and yesterday's open interest

G= the greater of Todays or yesterday's open interest

Herrick Payoff Index Formula ^ [31] (Achelis)

[31] (Achelis)

When the HRI is above “0” zero, it shows liquidity increasing in the futures contract representing a bullish movement. And on the contrary, when the HRI is below “0” zero, it shows money is leaving from the futures contract which would signal as a bearish market. The main focus of the HRI is to look for divergences between the price and index. Figure 20 shows the HRI indicator in use.

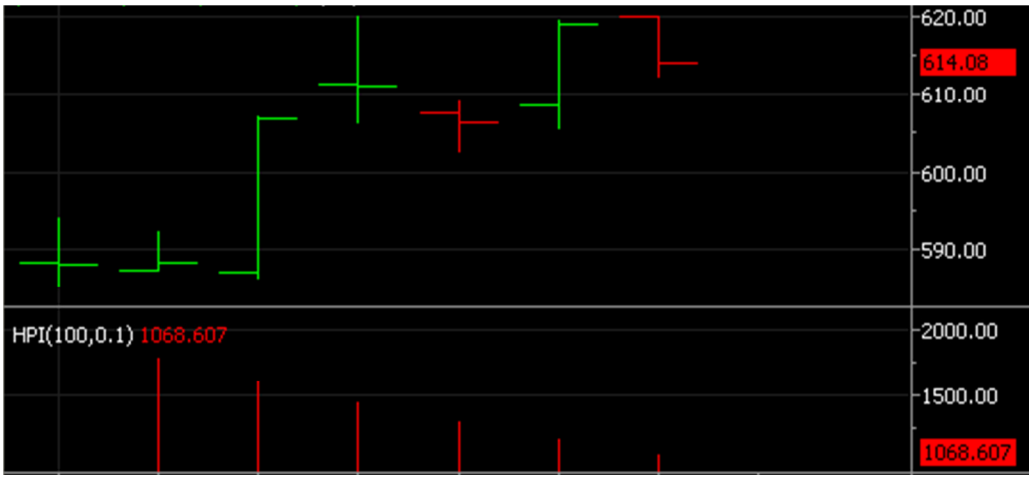


Figure 20: HRI

2.5.8 The Ultimate Oscillator

This indicator sums up the true ranges of the number of bars specified by the shortest, intermediate and longest average, all of these depending on the timeframe set by the user. These additions are then divided into the sums of the change between the close to the low and then plotted on the chart.

This is known as an oscillator that expresses momemtum across the three time frames listed above. This is the main difference between the UO and the other indicators out there working with the same principle. The ultimate oscillator attempts to correct the fault of a bearish divergance by incorporating longer time frames into its formula.

The UO formula is the following:

$$Close - Minimum\ (Low\ or\ Prior\ Close).$$

$$TR = Maximum\ (High\ or\ Prior\ Close) - Minimum\ (Low\ or\ Prior\ Close)$$

$$Average7 = (7 - period\ BP\ Sum) / (7 - period\ TR\ Sum)$$

$$Average14 = (14 - period\ BP\ Sum) / (14 - period\ TR\ Sum)$$

$$Average28 = (28 - period\ BP\ Sum) / (28 - period\ TR\ Sum)$$

$$UO = 100\ x\ [(4\ x\ Average7) + (2\ x\ Average14) + Average28]/(4 + 2 + 1)$$

This indicator is most commonly used on daily and wekly charts and parameters must me adjusted by the user for proper overbought and oversold readings. A Sell signal is formed in a bearish divergence and a buy signal is formed in a bullish divergence.

For the bullish divergence, the oscillating wave forms higher lows when the prices are pushed down and the higher low in the oscillator shows a loss in momentum. And for a bearish divergence, the oscillator makes a lower high as prices are pushed upwards to higher highs. Thus making the oscillating wave show a loss of upward momentum. The following picture shows the Ultimate Oscillator being implemented in the tradestation platform.

Figure 21: Ultimate Oscillator

2.5.9 Bollinger Bands

The Bollinger Bands, or just "bands", indicator was developed by John Bollinger and consists of three buffers on a chart that do not intersect. The center buffer is the calculated moving average (see **Moving Average**) for the chart. The upper and lower buffers are calculated by the moving average plus the standard deviation, and the moving average minus the standard deviation, respectively. Most commonly, the outside buffers are set two standard deviations from the moving average. This acts on the currency pair as a bell curve, where prices above the upper deviation buffer are expected to drop back towards the moving average, indicating to short, and prices below the lower deviation buffer are expected to rise towards the moving average, indicating to long. In addition, bands display the volatility of the market; a more volatile market causes a wider gap between the moving average buffer and deviation buffers, while a less volatile market generates a "tightening" of the bands. "The tightening of the bands is often used by technical traders as an early indication that the volatility is about to increase sharply."[32] (Definition - Bollinger Band)This indicator is most useful during a relatively static market as external events such as political changes can drastically change the market, invalidating a statistical analysis.

Calculations:

Middle Line $ML = \frac{\sum P_N}{N}$

Top Line $TL = ML + (D * StdDev)$

Bottom Line $BL = ML - (D * StdDev)$

Standard Deviation StdDev $= \sqrt{\frac{[\sum P_N - SMA]^2}{N}}$

P_N = *Price at period*

N = *number of Periods used*

SMA = *Simple Moving Average*

[32] (Definition - Bollinger Band)

2.5.10 MACD

Moving Average Convergence Divergence (MACD) momentum indicator that shows the relationship between two moving price averages that was developed by Gerald Appel in the 70s to predict prices and their strength. It allows both the buyer and seller to know who is driving the price in the market.

It is calculated by subtracting the 26-day exponential moving average (EMA) from the 12-day EMA. The plot then forms a line that fluctuates above and below zero without any upper or lower limits that means if the MACD is positive and rising, the gap between the 12-day EMA and 26-day EMA is widening. If this happens, it shows that the rate of change of the faster moving average is higher than the rate of change for the slower moving average, and this is considered Bullish as the momentum is positive. On other hand, if the MACD is negative and declining more, then the negative gap between the faster moving average and the slower moving average is expanding. This means that downward momentum is accelerating and this would be considered bearish.

The chart created below shows the MACD indicator at the bottom of the price chart

Figure 22: MACD indicator

There are three important ways that MACD is used to interpret the momentum and the trend.

The Crossover happens when the MACD crosses from above to below the signal line and this indicates a sell signal.

The second is the divergence and this happens when the MACD histogram is making a higher high and the prices is making a lower high.

The third is the overbought and oversold, this indicates when the market is overbought or oversold. In this case when the shorter –term MACD moving average separates or diverges strongly from the longer-term MACD moving average, it shows that markets are over-extended. If this happens it indicates that markets needs to correct or consolidate before moving forward with the trend.

Applying the three methods in interpreting MACD can be of great advantage to both the buyer and the seller in today's volatile market

2.5.11 Momentum Indicators

Momentum indicators are technical tools use in Trading to measure the rate of change in closing prices of stocks, commodity or currency pair. In forex trading, momentum indicators are used to track the weakness or latent strength in currency pair and also signal when potential turnaround is going to happen. Momentum indicator is usually laid out as a single line on its own chart separate from the price bars, and can always be located at the bottom part of the price chart. The indicator is drawn with a center line of zero and it can be calculated by subtracting the previous n-period closing price selected by the trader from the current close price.

$$\text{Momentum} = CP - nCP$$

Whenever the indicator line crosses beyond the center line, it generates a positive momentum reading and that indicates a bullish signal which means it is time to buy. In contrast, when the indicator line crosses below the center line, it generates negative readings which is a bearish signal and that means it is time to sell. The last signal is when the indicator line start to reverse towards the center line, it signals that it is time to exit the market otherwise all the profit made will be eroded. The three

momentum signals are indicated on the EURUSD chart below. An example of the Momentum indicator is shown below in the chart of the EURUSD currency pair.

Figure 23: Momentum indicator

(This Figure allows us to notice the momentum of the Dark Blue line as it passes the center red line)[33](Momentum)

2.5.12 Rate-of-Change (ROC)

Rate of Change (ROC) indicator is another technical analysis tool use to forecast the underlying trend in the market. It is used to measure the percentage change in price over specific time period to determine the direction of the markets. Using this indicator the trader can follow the rising market through watching the rate of change indicator reading. The rate of change indicator can be used to determine when to stay and leave the market index.

[33] (Momentum)

ROC signals included the Center crossover, divergence and the overbought –oversold readings; this shows the upward and downward movement. For example if the rate of change remains positive, the price will rise and if the rate of change decreases the price will fall.

$$ROC = (today's\ close - close\ n\ period\ ago)/(close\ n\ period\ ago) \times 100$$

ROC indicator are oscillating around zero, in this case the buying signal generated when the line of rate change crosses above zero line and selling signal is the opposite, that is when it crosses below zero.

Another way of using ROC is when the selling is generated the value of indicator begins to fall after the generated maximum and buying when it begins to grow after local minimum.

Observation of divergence can provide warning or alerts of the weaknesses in the market trend; however this does not represent the buy or sell signal. The divergence occurs when the ROC indicators trend goes in the opposite direction of the price trend.

Therefore is important oscillator that allows the trader to compare today's value to a previous, there by measuring changes in the market. It guides the trader on market future direction and speed. So ROC can guarantee trader the very moment the market shifts direction.

2.5.13 Stochastic Indicators

Stochastic indicator is in the family of momentum oscillators and it was developed by George Lane in the 1950s. Lane observed that as markets reach the peak, the closing price tend to approach the daily highs and lows. Basically, stochastic indicator is an indicator of upward and downward market trend.

The StockChart-Chartschool notes that according to an interview with Lane, the stochastic Oscillator does not follow volume or anything like it; it follows the speed or the momentum of price. He notes that the momentum changes directions before price, as the strength and weakness diverge in the stochastic oscillator and this can be used to predict the price reversal.

As noted by Thorp A. Wayne that Stochastic is a process involving random variables. It consists of two lines, %K and %D lines also called the fast and the slow. The %K compares the latest closing price to the recent trading range and the %D value is the 3- period MA derived from taking the moving average of the %K.

%K=100×(	$\text{Recent Close} - \text{Lowest Low (n)}/$ $Highest\ High - Lowest\ Low\ (n)$
%D	3-period moving average of %K
(n)	Number of period used in calculation

[34] (Thorp)

%K= (Close –Lo) ÷ (Hi –Lo) ×100

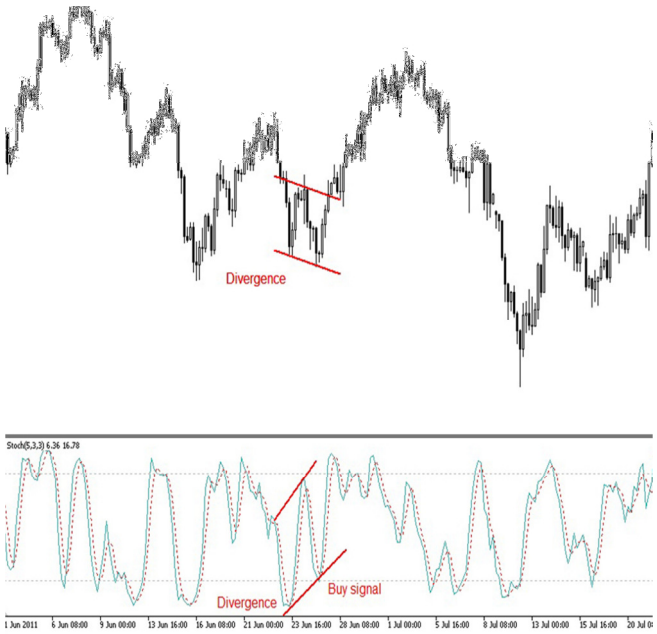
Close = Last closing price

Hi = Highest intraday price over the designated period

Lo = Lowest intraday price over the designated period

The two lines are plotted below the price chart and known as the %K and %D lines and the default setting for indicator is 14 time period. For example if the trader take one 1 hour chart, the closing price of the currency will be compared to it prices from 14 hours ago. As indicated below

[34] (Thorp)



These Figures show %K & %D

Figure 24: Stochastic Indicator

The stochastic Indicator is used to determine overbought and oversold levels in the market that is to determine when the upward and downward move is going to occur by looking at where the two lines cross. When the lines are above 80, it represents a market that is potentially overbought and when they are below the20 it indicates that the markets are potentially oversold. [35] (Lakhani)

The Second use is trading crossovers. The trader has to watch for when there is a crossover of the %K and %D. When %D is below the 20 mark and the faster %K line and the %D line, it is a sign that the market may be heading up and when %D is above the 80 mark and the %K line crosses below the %D line this is a sign that market may be heading down.

The other is the Trading stochastic divergence which happens between the stochastic and price can be used as a leading indicator for existing trading signals. In this case, if the prices are making new lows and stochastic moves higher or crosses to the upside and this can warn the trader that the prices may rebound as price move up. [36] (Stochastics Indicator)

[35] (Lakhani)
[36] (Stochastics Indicator)

2.6 Trading Platforms

A Trading platform is Software used by traders and Investors to manage, open and close positions in any market, whether it would be Stocks, Options, Futures or Forex. These Platforms often include software that allows traders to better analyze the market, and provide tools for individuals to develop their own analysis techniques.

2.6.1 TradeStation vs. Metatrader

Two trading platforms were recommended by our Project advisor, TradeStation and Metatrader. In order to choose a platform we went through the process of analyzing each platform and choosing one to develop our final indicator in.

Popularity

MetaTrader is the world's most popular trading platform, due to different variables. Some of these being a free platform, and having to freedom to access the latest data which is updated constantly on top of offering the trader a demo account with simulated money to put their strategies to a test. TradeStation, like MetaTrader, offers competitive pricing, Fast market data, account security, a World Class customer support and a large set of extremely helpful Built in analysis techniques.

MetaTrader's scripting language is MQL4, this language is "C" based and I used to write scripts which are used as indicators, and also allows for the development of Expert advisors, which is a system that works on parameters and conditions set by the trader, the get in and/or get out of the market.

TradeStation uses "EasyLanguage" which was created by the software developers, although it does not provide the trader with the freedom of MQL4, EasyLanguage can make complicated analysis in very few compact lines of code.

Main Differences

EasyLanguage code always executes on the close of each bar on the chart in which the desired strategy is being applied. A Strategy in TradeStation has the same purpose of an Expert Advisor in MetaTrader, This can be an issue for investors who do short time trading now that it can be crucial to

enter the market at a specific time frame, its effect on long term traders is not necessarily a big issue now that these hold their positions for longer time frames

On the other hand, MQL4 has a command that allows the trader to execute actions on each tick. This is very helpful for accuracy, and allows the individual to have more control over his position.

Executing a trade in EasyLanguage allows a user with a short position, to place a long position entry. If the position size of this entry is not specified the long entry automatically closes the short trade. On the other Hand with MQL4, the trader can be left with multiple competing positions, which need to be cancelled individually.

We decided to use MetaTrader for the fact that it gave us the broadest set of tools to develop our system the way we wanted it. Although TradeStation can be used for testing some indicators, and looking at smaller time frames. Metatrader was our platform of choice to develop our final product.

Chapter 3: Methodology

As the group kept working on the design of indicators that showed strength and overall pip change combined with an alerting system, the main focus became the construction of a universal indicator that would combine all of the previous and allow the user to modify settings according to their trading strategy. Thus the FXCM Multiple Currency Overview Indicator was created. With that the group also came up with a general trading strategy called the Three Stage Trading Strategy. Each group member then used the Indicators, the trading strategy and knowledge gained through research to practice trading. This Section will cover an overview of the final FXCM Multiple Currency Overview.

3.1 FXCM Multiple Currency Overview

3.1.1 Components

- **Chart**:
    - Shows all 8 major currencies, as well as an overall PIP change
    - Inverted Pairs to display each currency as a whole
    - Multiple time frames to see behavior of trends
    - Displays trade recommendation for each pair, based on window periodicity
    - displays the history of the strength of each currency, up to 24 bars ago
    - time frames timed accurately, with multiple bars calculated in each time frame
    - summarizes 224 windows on one screen

- **Strength**
    - user defined ratio of
        - number of pairs in favor of each currency
        - ranked pip change of the currency
    - Shows how well a currency has done over the past 24 bars
    - Allows support reference for trading recommendations

- **Moving Average Analysis:**
    - SMA to show general direction of currency
    - EMA1 & EMA2 to work with SMA in common trade patterns
    - Buffers for assurance of direction of market
    - Multiple Alerts per Bar, to keep track of the market
    - Alerts include order type, time, price, spread, and cause of the alert
    - Error checking to make sure no false alerts are created

Figure 25: FXCM Multiple Currency Overview

3.1.2 Moving Average Trading Conditions

- **Buy condition 1**
  - *price crosses above SMA*
  - *shortEMA is above longEMA*
- **Buy condition 2**
  - *shortEMA crosses above longEMA*
  - *price is above SMA*
- **Sell condition 1**
  - *price crosses under SMA*
  - *shortEMA is below longEMA*
- **Sell condition 2**
  - *shortEMA crosses under longEMA*
  - *price is below SMA*
- **Close condition 1**
  - *shortEMA crosses above longEMA*
  - *price is below SMA50*
- **Close condition 2**
  - *shortEMA crosses under longEMA*
  - *price is above SMA50*
- **Close condition 3**
  - *price crosses under SMA*
  - *shortEMA is above longEMA*
- **Close condition 4**
  - *price crosses above SMA*
  - *shortEMA is below longEMA*

Triple Moving Average Theory

The triple moving average treats the market roughly as a complex sine wave, using the SMA (default 50) as the median. Two exponential moving averages (default 5 and 20) act as a way to follow the slope of the sine wave, where a cross occurs when the slope has reversed. Buy condition 1 occurs when there is an upward slope that passes the median, implying the beginning of a phase from 0 degrees to 90 degrees. Close condition 2 occurs from 90 degrees to 180 degrees, where price is above the SMA and the slope has turned to retreat towards the median. Sell Condition 1 occurs from 180 degrees to 270 degrees, following the inverse of buy order 1. Close condition 1 follows 270 to 360 degrees, following the inverse of close position 2.

Following the concept of a complex wave, Buy condition 2 occurs if the price changes to include additional peaks before returning towards the median. Sell condition 2 behaves in the same manner for troughs below the median.

Close conditions 3 and 4 occur if the market is close to the median, where the price rises slightly above the median, indicating a buy order, then passes and continues below the median to indicate a sell order. These close conditions ensure that there is always a closing position between any two opening conditions.

3.1.3 FXCM Multiple Currency Overview External Variables

Display Settings

Column_width	Width of the cell columns for the graph displayed
Row_height	Height of the cell rows for the graph displayed
Column_offset	Distance from the side of the graph to the side of the screen
Row_offset	Distance from the top of the graph to the top edge of the screen
fontSize	Size of the text font
numberSize	Size of the numbers in the charts

Strength Settings

Rank_weight and *positives_weight* is a ratio in determining the strength of the currency, displayed in the bottom right of the window.

Rank_weight	Sums PIP change of the current bar for each pair of a currency. This is done for each of the eight currencies. The currencies are then ranked according to total PIP change, where the currency with the greatest positive PIP change receives rank 8, and the currency with the greatest PIP change receives rank 1.
Positives_weight	The number of currency pairs that are in each currencies' favor, from 0 to 7. This is scaled to a strength of 8, where all pairs are positive for a

currency, to a strength of 1, where all currencies are negative for a currency.

TitleColor Font color of the currency names for the strength graph

Number1 Font color of strength 1 for the strength graph

Number2 Font color of strength 2 for the strength graph

Number3 Font color of strength 3 for the strength graph

Number4 Font color of strength 4 for the strength graph

Number5 Font color of strength 5 for the strength graph

Number6 Font color of strength 6 for the strength graph

Number7 Font color of strength 7 for the strength graph

Number8 Font color of strength 8 for the strength graph

Moving Average Analysis Settings

SMA number of bars included in calculating the simple moving average of each currency pair.

EMA1 Number of bars included in calculating the first exponential moving average of each currency pair. Must be greater than 0

EMA2 Number of bars included in calculating the second exponential moving average of each currency pair. Must be greater than 0 and cannot be equal to EMA1.

ALERTS_PER_BAR Sets the limit to the number of alerts that the indicator will prompt within a given bar. This number is tracked independently for each currency pair. Setting *ALERTS_PER_BAR* to 0 tells the indicator that there is no limit.

WAIT_AFTER_ALERT	Sets the minimum amount of time, measured in seconds, between alerts. This is tracked independently for each currency pair. This prevents the indicator from spamming alerts when currency pairs hover around crosses.
BUY_CROSS_BUFFER	The number of PIPs that the currency has to reach beyond a cross in order to trigger a buy alert. Used as assurance of the direction of the currency
SELL_CROSS_BUFFER	The number of PIPs that the currency has to reach beyond a cross in order to trigger a sell alert. Used as assurance of the direction of the currency
CLOSE_CROSS_BUFFER	The number of PIPs that the currency has to reach beyond a cross in order to trigger a close order alert. Used as assurance of the direction of the currency
GAP_MARGIN_OF_ERROR	The margin of error (decimal) allowed in the history of a currency pair that is allowed in order to generate alerts for that currency. For example: With a *GAP_MARGIN_OF_ERROR* of 0.1 and SMA the highest moving average of 50, the indicator will generate alerts as long as there are not more than 5 bars missing from the history of the currency pair.
*Track_*******	These 28 Boolean variables indicate whether or not alerts will be generated for the corresponding currency. Allows the user to focus on only a few currencies if desired.
*Trade_on_***day*	Seven Boolean variables that indicate whether or not alerts will be generated during the corresponding day. Allows the user to specify a time of week to trade.
Sunday_Open…	The time on Sunday that the indicator will begin generating alerts. Each day has a corresponding opening time.
Sunday_Close…	The time of Sunday that the indicator will stop generating alerts. Each day has a corresponding close time.

3.1.4 FXCM Dynamic PIP Overview External Variables

ranked_by_column	Which column the currency pairs are being ranked in accordance to.
scaleX	horizontal interval at which the cells are created
scaleY	vertical interval
offsetX	horizontal indent of all cells
offsetY	vertical indent
fontSize	title font size
numberSize	data font size
history_shift	number of bars in the history that the indicator is calculating
fontColor	Default Color of font
titleFont	Default Title font
positive_color	Default color of positive PIP change
negative_color	Default color of negative PIP change

3.1.5 Display Setup and Functionality

Setting up the FXCM Multiple Currency Overview

The FXCM Multiple Currency Overview can be set up effectively in any window, but can update faster when attached to more active currency pairs. This is due to the indicator behavior of MetaTrader indicators, as the indicator will only update whenever information comes in for the window that the indicator is attached to. FXCM Multiple Currency Overview gets attached to the chart the same way as any other indicator, but requires additional one-time steps after opening the indicator.

Metatrader is not designed to display tables, and therefor there is no way to turn the existing chart off. However, the existing chart can be effectively hidden by implementing the following steps. Either right-click anywhere in the window and select “Properties”, or press “F8”. In the common tab, make sure that the “Show grid” and “Chart on foreground” options are not checked. Then, in the colors tab, change the color of all but the foreground to black. It is important to still be able to see the foreground, as there are comments reporting the running status of the indicator displayed in the top left of the screen. This should produce a black screen, with only the indicator objects displayed.

The first time that the FXCM Multiple Currency Overview is run, it will most likely alert the user of an invalid history warning. Under most circumstances, the indicator will eventually load all of the history required. However, if the complete history for a currency pair is missing, the user may need to open a new window for the currency pair and tab through each time frame. This process can be time consuming, but once the history for each currency pair has been loaded, the terminal remembers it on that computer and will be able to remain up-to-date automatically. The next time the indicator is opened and the history alerts prompt, it is due to the gap in time between when the program was closed to when it is opened again, in which case MetaTrader will be able to automatically gather the missing information within a few minutes.

Maintaining Timeframe Accuracy

Metatrader measures beginning of the bar as a constant time interval. E.g. 1:00, 1:05, 1:10. This creates inaccurate measurements of the current bar at times such as 1:01, 1:06, 1:11 because the information of the bar in that time frame is only measuring one minute, compared to the five minutes of previous, complete bars. This inaccurate measurement creates an outlier when calculating the average of PIP change over multiple bars, which can cause an indicator to fail to alert, or incorrectly alert, the user of a trading opportunity. In order to maintain reasonable precision and accuracy when calculating and displaying information, the FXCM Multiple Currency Overview takes advantage of smaller periodicities. The program contains a condition for each possible periodicity, in which the current bar's information is calculated by several smaller bars. For example, the five-minute periodicity is calculated using five one-minute bars instead of one 5-minute bar. Using this method, the five-minute periodicity would display the PIP change from 1:01 to 1:06, rather than 1:05 to 1:06.

3.2 Three Stage Trading Strategy

3.2.1 Stage One

FXCM Overview Indicators

The first stage of the trading strategy is finding out which currency pair the user wants to trade. This can be done using the FXCM Multiple Currency Overview indicator.

First in line is the moving average analysis. This will suggest to the user any currency that meets the trade conditions for the triple moving average algorithm. These trade orders are triggered under special circumstances and are always worth a look if an alert pops up.

Second, the Strength History provides a quick summary of which currency is doing the best, which currency is doing the worst, and everything in between. The history also allows the user to find patterns in the behavior of how a currency moves. For example, in smaller time frames, it is common for currencies that have a very high strength during one bar to have a very low strength in a following bar. This case is often due to a “rebound” in the market. Pairing high strength currencies with low strength currencies can help the user select the most volatile, and therefore most profitable, currency pair. In contrast, pairing two currencies that have consistent strengths between three and six can help the user select lower risk currency pairs.

Finally, the eight currency charts say exactly what is going on in each window for each currency. Each row of these charts show how the currency has behaved for a given time frame, while each column displays both long and short term behavior for the corresponding currency pair. The combination of these give a great perspective of the overall movement of the currency over the past month, allowing the user to decide whether a buy order or a sell order is more appropriate for a given currency.

If the user prefers a more simplified version, in which the 28 currency pairs are listed in numerical order from the largest PIP change to the smallest PIP change, then he or she can use the FXCM Dynamic PIP Overview indicator. This indicator does not contain the moving average analysis or the Strength History, but is a quick way to see which currency pairs are the most active.

3.2.2 Stage Two

Pattern Recognition and Confirmation

Once the user selects a currency pair to focus on, he or she should open the window for that pair and analyze its behavior within the desired time frame. Here, the user can decide whether or not to pursue a trade for that pair.

Pattern recognition is essential to making any decision. The most obvious pattern is a linear trend. If a currency has been consistently trending down, the trader may not want to execute a buy order without evidence of a change in direction. Less common, yet arguably more important, are bearish and bullish patterns. These are patterns of four or five bars that commonly signify that the market is going to go in a certain direction.

Sideways markets are generally unpredictable, and the window view is the easiest way to recognize if this is happening. This is very important to look out for as it is the main weakness for the triple moving average algorithm, where the user can be prompted to trade in a currency pair that is stagnant. Another scenario to be aware of is sudden jumps in the market. These are usually instigated by an event in the news, and the amount that the market will move is unpredictable, and very often rebounds. These are high-risk scenarios where the user can make, or lose, a lot of money

3.2.3 Stage Three

Never Make a Blind Trade

Whether there is a sudden fifty pip drop, or a gradual two hundred pip climb, there is often some reason for it in the news. If stages one and two are indicative of a successful trade, the trader should make sure that there is not something in the news that contradicts it. It is important to note that anything that happens in the news can already be seen in the market so, with practice, the trader can get an idea of the amplitude of the effect that different news has on the market. However, movements in the market do not necessarily have news directly corresponding to them. For example, if news comes that is harmful to the euro, the EURUSD pair usually drops. In the meantime, there is no news for either

the US dollar or the Australian dollar, but since EURUSD is moving in favor of the US dollar, there is a higher chance for traders to continue to trade in favor of the US dollar, even though there was no news for the AUDUSD pair. The news acts as a source of confirmation for the behavior of the market, as it is the primary driving force.

3.3 Implementation of Trading Strategy

3.3.1 Implementation of the Three Stage Trading Strategy by Mariela Qirici

Although the trading strategy consists of three stages, the order in which stages two and three are executed is interchangeable. The first stage of the trading strategy involves figuring out which currency pair to trade. The FXCM Multiple Currency Overview indicator was used to make the final decisions on the currency pairs traded. However, close attention was paid to the time of Day that the trading was being done, so the useful information gathered from background research on Trading Sessions could be used to help with the decision. For a better understanding of how this was gone and how I implemented the rest of the trading strategy I will continue by explaining in detail the process using an example.

It was early morning when Mariela was trading. The North American trading session was about to start and she didn't want to spend too much time in the market, so it would be best to trade the more active currency pairs at that time which would be pairs involving US and European countries, specifically EUR/USD, GBP/USD and USD/CHF. To figure out exactly which currency to trade, she looked at the FXCM Multiple Currency Overview Indicator and noticed negative pip changes for the Euro in multiple timeframes thus decided to open up the EUR/USD chart and put in two EMA indicators matching the EMA1 & EMA2 settings of the Moving Average Analysis part of the FXCM Multiple Currency Overview. This was done to get a better visual of what was going on. The market was sideways since at this time it was still early, before 8am (NY open), providing the perfect opportunity to check out the news and avoid a blind trade (Stage 3).

FXstreet.com was used to check out the Economic Calendar and see if any important news had been, or was going to be, released that morning. The next news on the econ calendar that would be released for the Euro zone at the time was the Manufacturing Purchasing Managers Index which

captures business conditions in the manufacturing sector. Since the Manufacturing sector dominates a large part of total GDP, the manufacturing PMI (Purchasing Managers Index) is an important indicator for business conditions and overall economy in the EUR zone. Usually results over 50 signal bullish for the EUR and below signal Bearish. With the consensus being at 48.5 and previous news of consumer confidence contracting we suspected that the actual PMI would be lower than the prediction, and even if it wasn't lower it would still be under 50 resulting in a downtrend of the EUR. (Economic Calendar) With that Info in mind, it would be best to short the EUR/USD according to Stage 3 of the strategy however the decision to do so would have to come from Stage One and Two.

Right before the NY open I started observing the candlesticks pattern. Throughout trading the patterns pictured below was used as guides on whether the market would go Bearish or Bullish. The Patters that have both star and plus (*+) markings above them signify an even stronger chance of the market going the specific direction then the ones with just a star (*).

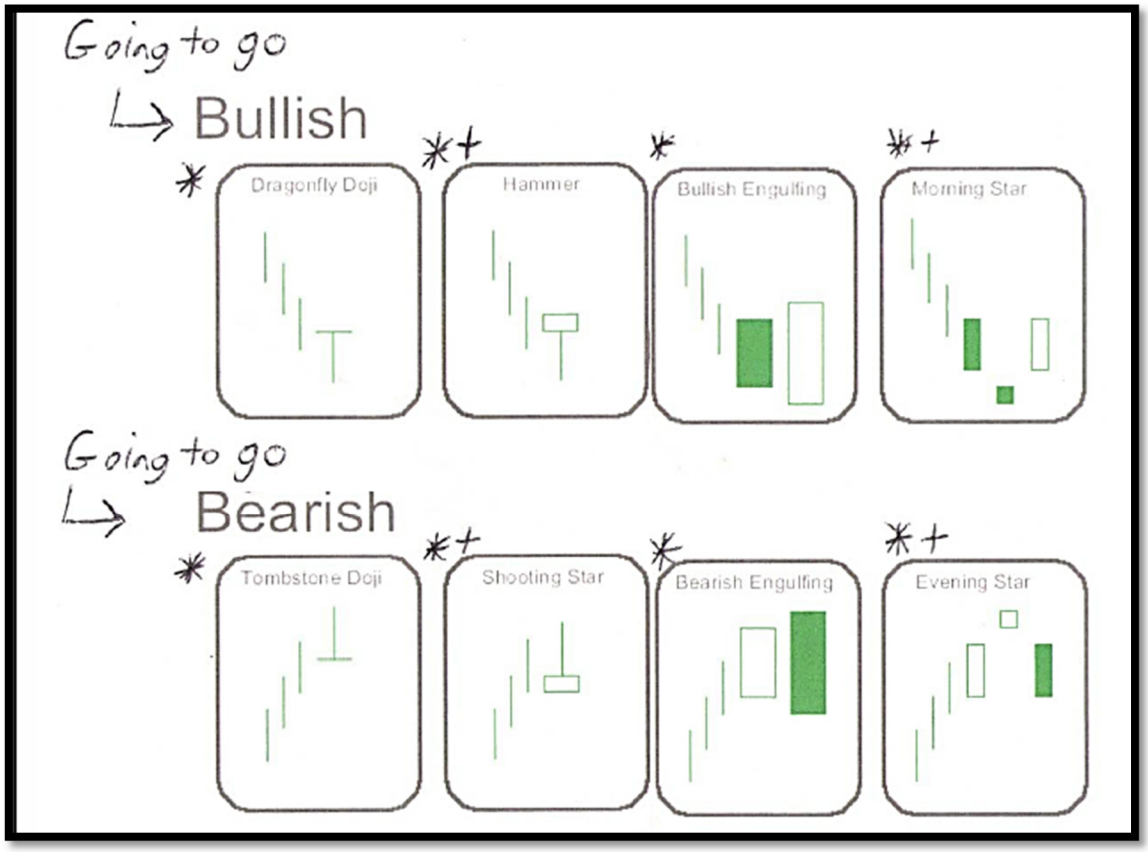


Figure 26: Candlestick Pattern

(Online trading Concepts – Candlestick pattern Recognition)

Just after the NY open, the candle stick pattern looked like a “Shooting Star” which strongly suggests the market would go Bearish however the Moving Averages hadn’t crossed yet. Shortly after the news previously mentioned got released, the short EMA (red) crossed the long EMA (purple), which confirmed that a Sell order should be pit in (Stage 1). Mariela entered the market with the sell order at this time and exited after witnessing a “Hammer” candlestick pattern, making substantial profit. The graph below shows the drop in euro during the time of my trade and the small screen shot of the news released by FXStreet.com economic calendar. [37] (Economic Calendar)

Figure 27: Market Behavior- Implementation of 3 Stage Strategy Example

[37] (Economic Calendar)

3.3.2 Individual Trading Plans

As mentioned in the background and emphasized in lectures for this project, any trading system, whether based on fundamental or technical analysis (or a combination of both), whether auto-traded or manually traded, must possess an objective, a particular financial instrument (market) to trade, a particular time frame over which trades will be made, Entry and Exit Rules as well as Risk Management, System Monitoring Techniques and Asset Allocation Rules. These components must also be thoroughly tested so that they work together harmoniously to generate profits. (RRRR)

Given the focus of this project became the creation of indicators, specifically the FXCM Multiple Currency Overview a rigid trading system wasn't created. The group decided to personalize the project objective into creating a wonderful tool for any new trader to get started with forex trading, providing the reader with the tools to succeed by learning the basics through the background research provided, and then using the indicators created and the general three stage trading strategy according to one's own personality and goals. One group member focused on the coding and refining of the indicators created while rough trading systems were created by the other three group members and trading was done in order to test the effectiveness of the indicators. Each group member was given a $100000 investment capital.

Mariela's Trading Plan

Mariela's goal was to implement the three stage trading strategy best as she could and have high returns, with low time commitment for trading and without holding trades overnight. She used the Entry and Exit rules specified in the "Moving Average Trading Conditions" however she often changed the numerical values (number of bars included in the calculation) of the moving averages. The act of doing so gave leeway to act on instinct and not follow rigid rules. Instinct comes with emotions and emotions are bad for trading. When a lot of time was spent setting up and preparing for a trading session as explained in the above section, Mariela's trading was usually profitable. Trying to satisfy the low time commitment objective of the system she often let her emotions and impatience take over thus putting in a lot of trades when in contrary she should have thought carefully of every move she made. Most of time small timeframes were utilized, opening 5minute and 15minute charts during trading while observing the behavior of the larger timeframes through the FXCM Multiple Currency Overview Chart.

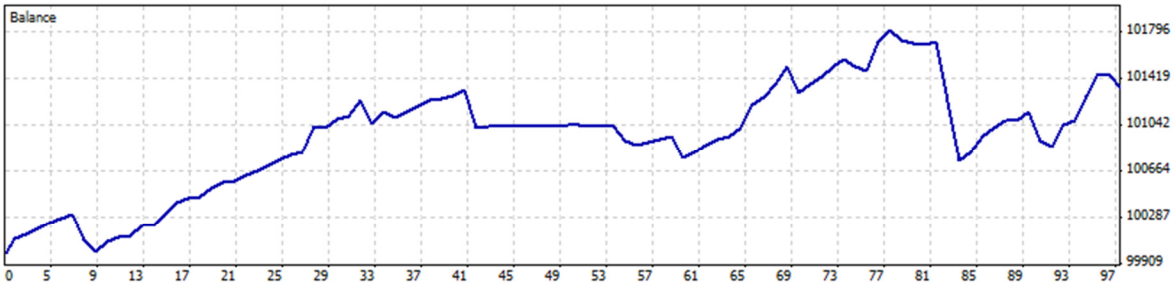
The FXCM Multiple Currency Overview Chart was also used to try and trade different currency pairs thus not depending on the behavior of a sole currency pair. Her risk management technique was to put 30pips stop losses or trailing stop loss of 25pips, combined with the restriction of trading a maximum of 5% capital per trade. The following is a summary of Mariela's latest trading, which reflects profit from this trading system, however the fact that this is not a system with rigid rules means that it is very risky and not a properly formulated trading system.

Trading Summary (Trading Record included in Attachment I)

Summary:

Deposit/Withdrawal:	**100 009.00**	**Credit Facility:**	**0.00**		
Closed Trade P/L:	**1 333.98**	**Floating P/L:**	**-164.73**	**Margin:**	**12 600.00**
Balance:	**101 342.98**	**Equity:**	**101 178.25**	**Free Margin:**	**88 578.25**

Details:



Gross Profit:	**4 332.73**	**Gross Loss:**	**2 998.75**	**Total Net Profit:**	**1 333.98**
Profit Factor:	**1.44**	**Expected Payoff:**	**15.69**		
Absolute Drawdown:	**0.00**	**Maximal Drawdown:**	**1 065.53 (1.05%)**	**Relative Drawdown:**	**1.05% (1 065.53)**
Total Trades:	**85**	**Short Positions (won %):**	**0 (0.00%)**	**Long Positions (won %):**	**85 (75.29%)**
		Profit Trades (% of total):	**64 (75.29%)**	**Loss trades (% of total):**	**21 (24.71%)**
Largest		**profit trade:**	**234.29**	**loss trade:**	**-485.86**
Average		**profit trade:**	**67.70**	**loss trade:**	**-142.80**
Maximum		**consecutive wins ($):**	**19 (1 020.14)**	**consecutive losses ($):**	**3 (-455.24)**
Maximal		**consecutive profit (count):**	**1 020.14 (19)**	**consecutive loss (count):**	**-968.85 (2)**
Average		**consecutive wins:**	**5**	**consecutive losses:**	**2**

Sebastian's Trading Plan

Sebastian's initial goal consisted on generating a profit near 5% by placing low risk trades and progressively increasing my Lot size. But most important of all he wanted to make this project a fun experience awhile also developing a competitive aspect to it for motivation within the group. It was decided to trade multiple currencies that had a low spread (which in turn symbolizes liquidity and small volatility) and also currency pairs, whose rate of change was very high during the 1 minute, 5 minute intervals now that these can indicate quick changes, short term rebounds and reversals in smaller time frame charts.

Before Jumping into the market, Sebastian would initially set up his trading platform with two windows, one consisted of the "XXX / XXX" desired currency pair with the according triple moving averages and a spread indicator located on the top right, the second window consisted of the dynamic pip overview which was used to identify trends and short term rapid PIP changes in the market. Secondly he looked at the economic calendar provided by The Forex Factory to keep track of the economic reports released earlier that day, and also future releases. (Forex Factory) On some occasions a small Yahoo! Forex news internet window was kept open to provide a continuous stream of the latest news. (Yahoo News)

The "1" and "5" minute time frames where used to identify fast movements in the market , the "15" minute time frame was mostly used for order placing and trend identifying, and the "1" hour – "4" hour time frames where used to predict future trends for the smaller time periods

All of the Buy/sell Conditions established in our system where followed closely when the opportunity arose.

- **Buy condition 1:** The current price crosses above simple moving average and the short Exponential moving average is above the long exponential moving average

- **Buy condition 2:** The short exponential moving average crosses above long exponential moving average and the current price is above simple moving average

- **Sell condition 1:** The Current price crosses under the simple moving average and the short exponential moving average is below long exponential moving average

- **Sell condition 2:** The short exponential moving average crosses under the long exponential moving average and the Current price is below simple moving average.

On the occasions when none of these conditions were met, the focus would switch to his other trading methods.

- **Trading Economic reports:** Sebastian would wait for a High importance economic report to be released, and as common sense would tell you, Positive values increased the value of the respecting currency and negative values would drop the currency pair.
- **Dynamic PiP Overview :** The Dynamic Pip Overview would point out rapid market changes , A good example of this would be the following: When the 1 or 5 minute time frames showed a big drop in pips (Red Color) but the 10 MIN, 15 MIN , 30 MIN, 1 HR , 2 HR , 4HR , 1 DAY shows a positive (Green Value) this indicates that there could either be an awaiting correction in the price to follow the general positive trend, or that there is a reversal taking place. The currency chart for that specific pair was opened to receive a better visual interpretation of the markets behavior and a decision was made, on most cases it would indicate an approaching correction in prize and in our example a BUY order would have been placed.

Risk management was mainly derived from looking at previous peaks, floors and ceilings in the market which pointed out the support and resistance levels, most trades were made between the current support/resistance levels to avoid an unexpected breakthrough. Some times as the market is trending strongly in one direction, random drops can occur, which causes many traders to hit their stop loss and then a few seconds or minutes later the market follows the previous trend. Although stop losses were not set manually, every trade was closely analyzed the entire time it was open, and my stop loss would just be set visually. 15 minute time frames had an average stop loss of 15 pips, 30 minute time frames had an average stop loss of about 21 pips. Position Sizing also played a big role in risk management, trades which carried uncertainty had smaller lot sizes, and the trades which promised an extremely low risk of loss had much larger lot sizes.

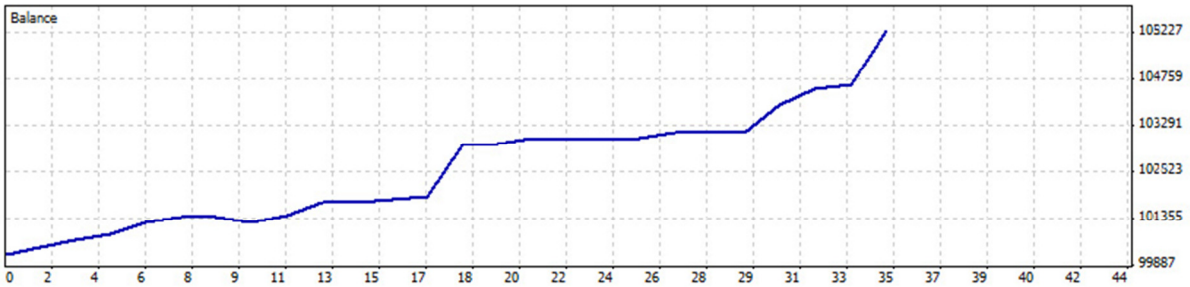
It was noted that the Dynamic PiP overview in combination with the spread calculation of (Ask – Bid), worked best in almost all environments, it provided a quick visual representation of market behavior across all currency pairs for easy identification of fast changing prices, this combined with low spread, allowed for profits the be taken at a faster and more efficient manner.

Trading Summary (Trading Record included in Attachment I)

Summary:

Deposit/Withdrawal:	99 996.30	**Credit Facility:**	0.00		
Closed Trade P/L:	5 252.24	**Floating P/L:**	0.00	**Margin:**	00.00
Balance:	105 248.54	**Equity:**	105 248.54	**Free Margin:**	105 248.54

Details:



Gross Profit:	5 319.35	**Gross Loss:**	67.11	**Total Net Profit:**	5 252.24
Profit Factor:	79.26	**Expected Payoff:**	187.58		
Absolute Drawdown:	0.00	**Maximal Drawdown:**	67.11 (0.06%)	**Relative Drawdown:**	0.06% (67.11)

Total Trades:	28	**Short Positions (won %):**	8 (87.50%)	**Long Positions (won %):**	20 (100.00%)
		Profit Trades (% of total):	27 (96.43%)	**Loss trades (% of total):**	1 (3.57%)
Largest		**profit trade:**	1 651.51	**loss trade:**	-67.11
Average		**profit trade:**	197.01	**loss trade:**	-67.11
Maximum		**consecutive wins ($):**	21 (4 936.97)	**consecutive losses ($):**	1 (-67.11)
Maximal		**consecutive profit (count):**	4 936.97 (21)	**consecutive loss (count):**	-67.11 (1)
Average		**consecutive wins:**	14	**consecutive losses:**	1

Jonathan's Trading Plan

The goal of this plan was to reach the targeted 5% profit per month without risking a large percentage of the trader's available capital. Six major currency pairs were traded during the Asian, London and New York session, from 10:30pm to 12:30am and 4am to 10am.Before trading a review of all fundamental information and any news item with the potential of affecting the market was reviewed. On top of that the multiple indicators created by the group were opened and used to help get a better understanding of the state of the market. The 15-minutes and the 1 hour charts were utilized to execute trades, since Jonathan found from experience that those timeframes were best suited for his trading. The 15 minute chart helped to enter and exit the market while the hourly chart enabled him to detect large fluctuations.

Jonathan used a combination of the group indicators combined with some substitutions to fit his personality and trading objective. The following settings were used for the moving averages: 5 EMA, 20 EMA and 50 SMA to trade. The buy conditions mentioned in the "Moving Average Trading Conditions" under the FXCM Multiple Currency Overview Indicator were used to enter the market. Jonathan decided not to wait for the indicators alerts to exit a position thus omitting the sell conditions mentioned in the "Moving Average Trading Conditions" sections. A combination of risk management, pattern recognition and a Support and Resistance indicator was used instead. The following is an example of how he used risk management to exit a position: if 10 pips were made in one trade he immediately exited the market. To exit based on pattern recognition the "going to go Bearish" patterns pictured under "Implementation of the Three Stage Trading Strategy by Mariela Qirici" were used. The three moving averages alone appeared to be very vulnerable in a ranging market. The graph pictured below shows how the moving averages can be vulnerable thus to exit a position entered during the quick uptrend Jonathan paid close attention to the patterns and when the "shooting star" looking bar appeared he exited the position. He integrated the above system with Support and resistance to help identify an exit position before the market reversed. He also avoided opening positions when the crossover is near his support or resistance.

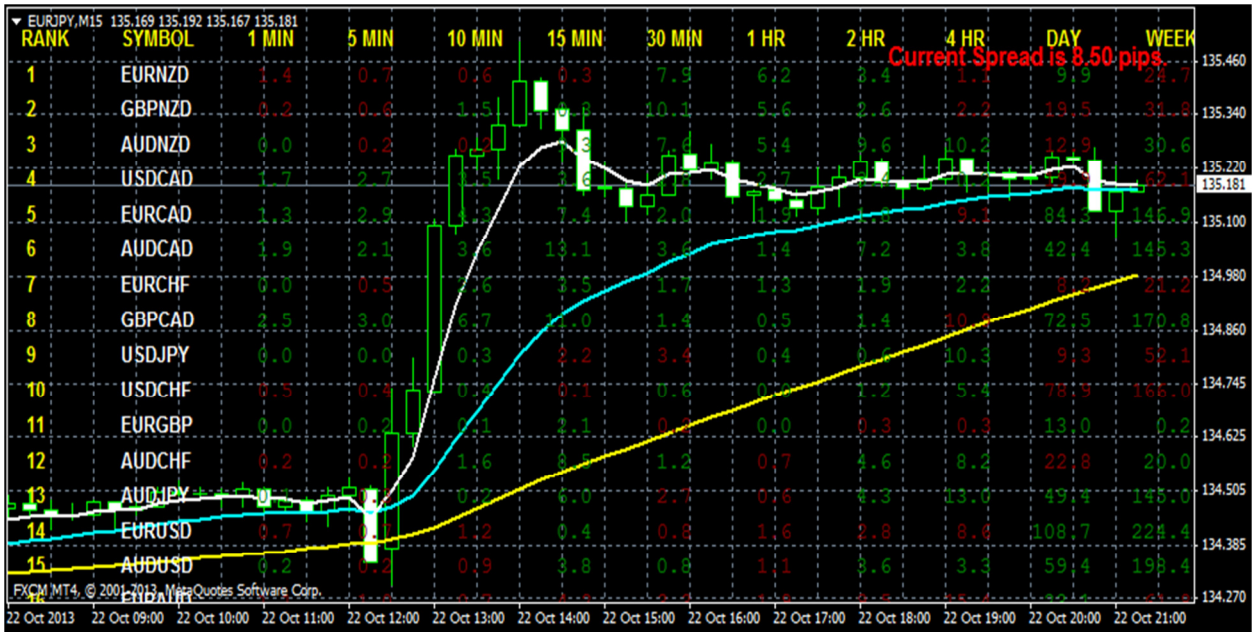


Figure 28: Vulnerable Moving Averages

For Risk management, Stop Loss positions were set at 20 pips below the entry point of any trade and Take Profit position was set at 50 pips above the spread. Since trading was done manually and careful observation of the marked behavior was taking place once in the market Jonathan decided to change the stop loss and TakeProfit when noticing a profitable trade. Trailing stop losses proved to be quite beneficial. No more than 5% of the account balance was risked at one time. Position size was only increased if a trade seemed profitable thus risk was minimized. When a trade was profitable Jonathan made sure not to be greedy and exited the market before it reversed.

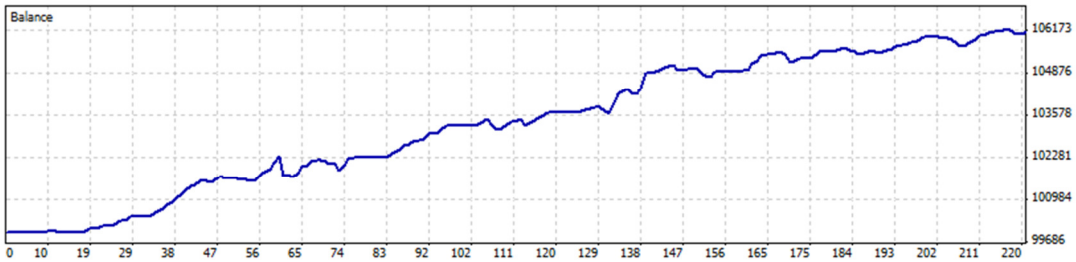
During the implementation of this system it was noted that Stop Losses set at 20 pips sometimes contributed to losses while trading due to market fluctuations which coincides with Mariela's experience, and is the reasoning behind her decision to set a larger stop loss.

Trading Summary (Trading Record included in Attachment I)

Summary:

Deposit/Withdrawal:	99 907.19	**Credit Facility:**	0.00		
Closed Trade P/L:	6 246.58	**Floating P/L:**	-310.00	**Margin:**	12 800.00
Balance:	106 153.77	**Equity:**	105 843.77	**Free Margin:**	93 043.77

Details:



Gross Profit:	9 896.49	**Gross Loss:**	3 649.91	**Total Net Profit:**	6 246.58
Profit Factor:	2.71	**Expected Payoff:**	34.13		
Absolute Drawdown:	1.63	**Maximal Drawdown:**	644.76 (0.63%)	**Relative Drawdown:**	0.63% (644.76)

Total Trades:	183	**Short Positions (won %):**	1 (0.00%)	**Long Positions (won %):**	182 (79.67%)
		Profit Trades (% of total):	145 (79.23%)	**Loss trades (% of total):**	38 (20.77%)
Largest		**profit trade:**	480.98	**loss trade:**	-613.15
Average		**profit trade:**	68.25	**loss trade:**	-96.05
Maximum		**consecutive wins ($):**	22 (1 612.97)	**consecutive losses ($):**	3 (-214.55)
Maximal		**consecutive profit (count):**	1 612.97 (22)	**consecutive loss (count):**	-613.15 (1)
Average		**consecutive wins:**	6	**consecutive losses:**	2

Chapter 4: Forex as a Business

As mentioned in the introduction, the goal of this project is to gather information and knowledge through research and use it to develop a winning trading system as well as to provide a detailed but simple approach of starting a forex trading business. Gaining in-depth knowledge and experience helps to build a profitable trading business thus trading beginners should acquire some experience in the field, research and learn as much as possible before considering starting a business. In the initial stages of creating a business one has to first trade with personal funds in order to build a good portfolio and then trade with other people's money. Throughout this project the group members traded and learned as much as possible about individual FOREX trading and discussed the possibility of creating a trading business and what needed to be taken in consideration to do so.

Type of business entity to register, what type of license required, and how the business can be incorporated, were things that needed to be researched. The group members also looked at the different types of tax policies, rules and regulations that would affect the trading business. Research showed that every country and state has different rules, regulations and tax laws that govern the forex industry. With that in mind, group members and the project advisor decided to focus on hypothetically setting up a trading business in the state of Massachusetts since that's where each member is located.

.

4.1 Types of Business Entities

In starting a new business, the business owner must decide what type of legal entity they would chose to register their business. This is very important decision for tax purposes. Business owners have several options in choosing an entity. There are five most common business entities: Sole proprietorship, Limited partnership, Limited Liability Company, S Corporation and C Corporation. A legal entity as defined by Jim Forrester is an organization recognized by the Internal Revenue Service by its corresponding employer identification number, whether it has employees or not. [38] (Forrester)

[38] (Forrester, 2005-10)

4.1.1 Sole proprietorship

Sole proprietorship is a simple form of business entity, created and run by one person as the owner of the business. It is usually unincorporated, and there is no distinction between the business and the owner. The owner of the business bears all the benefits and losses of the business. They do not own separate business account. As defined by Piper Mike sole proprietorship is referred to as "pass-through entity".[39] (Piper 13) Meaning that sole proprietorship is not taxed on its income, the business taxable income is passed to the owner.

The advantage of running a business as sole proprietor is that it is easy to start. There are no complicated legal documents to file and the startup cost is not high. A sole proprietor only needs to register his or her name to secure a business license and be ready to engage in business. More so, the business owner completely controls his or her business, the decision making process in regards to the business handlings are made by the owner. In addition the transferability of the business is solely prerogative of the owner.

However, sole proprietorship business owners suffer several disadvantages as well. The owner bears all the losses that the business incurs. Here the owner is not separate from the business; all legal liabilities like accident, debts etc. that the business incurs are borne by the owner. The liability does not stop in the business losses, but also affects the personal properties of the sole proprietor.

A Sole Proprietor is limited in investment funds which can sometimes hinder the growth of the business which would otherwise avert some losses. As stated by Forrester, when it comes to the nature of the trader's tax status, it is unclear what tax advantages a trader can enjoy, as the tax court laws continue to evolve. Forrester also adds that IRS does not trust sole proprietorship as it does with registered entity. Mistrust rises from the fact that during tax filing, sole proprietors do not select mark-to market accounting and so it becomes confusing in terms of reporting income on a scheduled D as capital gain and expenses on C as ordinary income. (Forrester)

4.1.2 Business Partnership

Business Partnership is formed by two or more persons who decide to pull their resources together to form a business enterprise. In a partnership, both profits and losses are shared equally by the partners as specified in the uniformed partnership act (UPA).[40] (Kubasek, Brennan and Browne 437)

[39] (Piper 13)
[40] (Kubasek, Brennan and Browne 437)

In a partnership, as with sole proprietorship, taxes are only levied on the partners personal income. The partners must report the income,deductions, gains and losses from its operation, but does not pay for income tax. Instead, a partnership passes-through any profit and losses to its partrners. However, partners must included their share of the partnership's income and losses in their tax return.[41] (IRS-Partnerships)

Business Partnerhip is divided in two parts, general partnership and limited partnership.

General partnership

A General partnership is formed when two or more persons pull together their resources to creates a business, share profit and losses. As indicated by Kubasek, partners share all unlimited liabilites like losses and debt. Partners can be sued by the creditors if the business incurs any liabiluty. More so, both their personal property can be used to settle the creditors. In Gerneral Partnership, Partners share the responsibilities partaining to the business.[42] (Kubasek, Brennan and Browne) When talking about the difference between a partnership and a limited partnership Chriss Hinson clearly describes that in a general partnership "all partners have the ability to actively manage and control the business", that includes making decisions on how the business is run. [43] (Hinson)

Limited Partnership

A Limited Partnership is formed with one general partner that incurs unlimited liabilty and one limited partner who is only liable to the extent of his or her investment in the business. As stated on QuickMBA , the limited partnerships are the creatures of statute since they must file with the with the state to form the busines enterprise. The statute is the Uniform Limited Partnership Act(ULPA) created in 1916, revised into RUALPA in 1976 and ammeded in 1985. Unlike general partnership, in limited Partnership, the general partners are responsible for decision making, and day-to-day runing of the business. They also bear most if not all the unlimited liability, that includes all the losses and debt incured by the comapany. More so, in the Limited Partnership, if the properties of the partners are inadquate to to settle the partnership creditors, then the personal assets of the individual partner that bears the unlimited responsibility may be affected to settle the debt. [44](Limited Partnership)

4.1.3 Limited Liability Company (LLC)

An LLC is a statutory creation with its own separate legal identity, and it is a hybrid form of business organization that provides its members with both the liability of the corporation and the

[41] (IRS-Partnerships)
[42] (Kubasek, Brennan and Browne)
[43] (Hinson)
[44] (Limited Partnership)

operational flexibility and the pass through tax treatment of the partnership. [45] (Cody, Hopkins and Perlman 17) Just like business partnerships , it can be formed by two or more people. However, the partners or members are not affected by the debt liabilities that arise from the business losses. The corporation, not the individual members, personal assets are used to settle business liabiliteis. In an LLC, members share in the profit from the business and exercise management control without these actions affecting their profit or limited liability status. [46](Kubasek, Brennan and Browne 469)

Limited Liability company member's Income, losses, deductionsand credits pass through the entity and are reflected on the retruns of the members, so the members must report profits and losses on their personal federal tax retrun. Another advantage that people who invest in limited liability company is that foreign investors can share the same benefits and limitations as the national of where a business is operating.[47] (Cody, Hopkins and Perlman 18)

However, Limited liability company have disadvantages worth mentioning. As outlined by Kubasek, states laws are not uniform, thus making complicated to figure out whether the state the business is operating from containes laws that apply to limited liability company. In Limited Liability company, decision making lies in hands of all the members, and so it can take time, resources and law suits by the members that are uncertificified by the other members decision. (Kubasek, Brennan and Browne)

4.1.4 C Corporation

C Corporation is independent legal entity that takes a life of its own apart from the shareholders. It can own properties, conduct business, and can sue and be sued.[48] (Corporation Types) Unlike other entities, C corporation shoreholders are not liable for the debt or obligations that arises from the business operations. As Stated by Hatten Timothy in his book *Small Business Management:Entrepreneurship and Beyond ,* in a C corporation, the most you stand to lose is the amount you have invested in it and that if the business fails or gets sued, your personal property remains protected from the creditors.[49] (Hatten 43)

[45] (Cody, Hopkins and Perlman 17)
[46] (Kubasek, Brennan and Browne 469)
[47] (Cody, Hopkins and Perlman 18)
[48] (Corporation Types)
[49] (Hatten 43)

The C corporation also files its own tax returns and pays income tax. The shareholders, however, have to pay taxes on their dividents. As it is put in the book *Contemporay tax practice: Research, Planning and Strategies*, the major disadvantage of a C corporation is the possibility of double taxation.[50] (Everett, Hennig and Nichols 15-41) Another disdvantage of a C corporation is the cost and the time it takes to register. The legal and administratives cost incured in starting a c corporation can be sizable disadvantage. (Hatten) The owner or the owners will have to consult with the legal aid, do paper work and register and all this takes money and time.

S corporation is a subset of a C Corporation. It was created to accommodate the small businesses. It was created to pass corporate losses, deductions and credit through to their shareholders for the federal tax purpose. The shareholders report the flow of income and losses on their personal tax returns and assess tax at their individual income tax rates; this therefore allows the S corporation to avoid double taxation on the corporate income. Unlike, C corporation, the S corporation, taxes are passed through to the shareholders, and still enjoy the limited liability, for example when the corporation incures losses and faces legal action or in terms of settling the creditors, the corporations assets are used, instead of the individual owners property. This may sound like a great option, however, there are great limitations to S corporation. One of the limitation as Stated by Schenk Deborah entitled Federal Taxation of S corporations, that a S corporation must file an election which requres affirmive action. In addition, the S corporation can have limited number of shareholders, and certain individuals, entities, estates and tax exempt entities may be eligible shareholders. [51] (Schenk 20) Since the individual owners are taxed not the corporation, they are taxed even when they have not received any divident. More so the S corporation is more complex in nature, more time take and the to form S corporation is high.

Having looked at the five different entities, It was concluded that the execution of a business needs to be placed as Limited Liability company. The main factors considered in chosing the entity are taxation treatment, Legal, personal liability, Flexiblilty, simplicity issues and cost of formation.

[50] (Everett, Hennig and Nichols 15-41)
[51] (Schenk 20.)

4.2 Tax Implications

Taxes are very important in forex trading and understanding the tax laws and system can be of great benefit to the trader. It is also important to know that tax laws are always changing and traders should always abreast with the frequent changes in the codes. Due to the frequent changes in the tax code, it most of the time leaves the traders confused on how to fill out their tax forms. In order to avoid tax confusion the traders should seek the advice of tax consultants. Like laws, rules and regulation, different countries formulate their own tax regulations that govern currency trading. So a trader should be aware of the tax laws in the country they are trading. According the IRS regulation, US traders who decide to move to another country and continue trading activities still have to pay the US taxes. For the traders in the United States, Internal Revenue Service (IRS) is the main institution that regulates taxes. Forex trading is taxed according to the capital gain and losses.

As described by the TradeKing Group, capital gains are generated when a trader earns a profit from selling a security for more money than they paid for it or buying a security for less money than they received when selling short. Individual traders and investors pay taxes on capital gains. Capital losses are generated when a trader incurs losses when selling a security for less than a trader paid for, that mean buying more and selling less. If a trader experiences capital loss they could deduct those losses up to the amount of gains they earned in one year but they can only claim up $3,000 if the losses are more than $3000 then they can moved it to another year.[52] (TradeKing Group)

Moving the losses depend on which IRS section contract a trader may chose. The two sections are 1256 contract and 988 contracts. Section 1256 contract, provides a 60/40 tax treatment. This treatment is explained by ForexOptions as follows: “Section 1256 applies to regulated futures contracts and it provides that no matter what a taxpayer’s holding period for a futures position is, 60% of any gain recognized is treated as long term capital gain and 40% of any gain is treated as short term capital gain”. Section 1256 contract is advantageous in that 60/40 rule allows a trader to blend 23% federal income tax rates to any gain.[53](ForexOptions)

On the other hand section 988 contract, gains and losses are treated as ordinary income or losses and so is subject to ordinary tax rates. In this case the individual trader has ability to claim capital losses as an income tax deduction, which means capital gains can be deductible 35% federal income tax. Lessambo in his book *The International Banking System; Capital Adequacy, Core Businesses and Risk*

[52] (TradeKing Group)
[53] (ForexOptions)

Management, states that "the taxation of spot forex is one of the opaque areas in the taxation of the financial products". He adds that Section 988 is the only section that provides for taxing for spot forex and "under this section gains and losses resulting from spot forex trades are treated as ordinary gains and loses". Forex trader with excess ordinary losses would not be allowed to carry back or carry forward its losses forward unless the trader is a professional trader. [54] (Lessambo)

Forward forex are subject to section 988 but a trader can opt to elect section 1256 contract. Section 1256 was created in 1982 by the congress to allow banks and large companies that trade in major currencies to opt for, however as stated by Lessambo must be able to deliver on three elements. That is the "contract must require delivery of the foreign currency, the foreign currency should be the one traded through RFCs and the contract must provide a settlement date. Taxation of futures forex is traded in an organized market with a clearing mechanism is standardized and offers mark to market options. Futures forex contracts are recognized foreign currency contracts subject to the preferential 60/40 rules under IRC section 1256, because futures forex are bilateral, traded in organized market and always come with settlement date." (Lessambo)

It is important to understand that treating forex activity under section 1256 is preferable when trading is profitable since it reduces overall tax on capital gains but if the trading activity results in a net loss section 988 is preferable.[55] (Forex Taxes)For a beginner trader starting a trading business, the leverage might be limited and the business may not have the financial capacity as the Interbank and big companies that trade in major currencies have. So with that in mind it would be wise to remain in section 988.

[54] (Lessambo)
[55] (Forex Taxes)

4.3 Forex Trading Regulations

Forex trading is one of the most fluid and volatile markets, open 24 hours, and with the evolution of technology, trading can be done from anywhere at any time. It is one of the largest markets in the world where millions of dollars get traded every day between banks, governments, big business, individual's investors and traders. With the vast variety of participants and money involved regulations are needed and do exist. These laws, regulations and rules have been formulated to guard, guide and protect the forex exchange market and to avert and eliminate the abuse of the traders and Investors thus it is very important for every trader to research and understand them. Forex trading is done through exchange of money from different countries, each country's government has formulated agency that responsible for overseeing the trading system to prevent fraud from within and without. This subsection will give an introduction to some important regulatory bodies the United States has to protect its traders and investors.

The National Future Association (NFA), this is a self-regulatory organization for the US futures industry. It safeguards market integrity and protect investors by implementing forex regulations. Membership here is mandatory for any future or forex broker operating in the US. As stated by (Maeda and Burrel), NFA establishes the rules that govern the forex markets and convinces the companies or industries to register with the organization and abide by established rules. They also keep the list of companies that have been disciplined for practicing sales fraud. Maeda and Burrel add that NFA permanently bars the companies that have been disciplined from the market engagement in their deceptive activities. [56] (Maeda and Burrel)

Another prominent regulatory body involved in forex trading is Commodity Futures Trading Committee (CFTC) which was created by congress in 1974 as independent agency with the mandate to issue forex regulation for financial markets in the US. It assures the economic utility of the market by encouraging their competitiveness and efficiency and protects market participants against abusive trading practices. As stated by Maeda and Burrel CFTC is limited in regulating the retail, over the counter forex markets in the US, but of recent has been granted the authority to regulate the sales of retail, over the counter forex futures and option but only for the regulated financial entities such as broker -dealer,

[56] (Maeda and Burrel)

Futures Commission Merchants (FCMs) banks and financial institutions, and has the authority to null unregulated forex entities.[57] (Maeda and Burrel)

Security and Exchange Commission (SEC) is section in CFTC and NFA that was created to protects the investors, maintain fair, efficient market and facilitates the formation of capital to sustain economic growth. It is charged with the responsibility of overseeing brokers and dealers, mutual funds, exchanges and investment advisers.

These regulatory bodies according to Globe Gain are judged with the responsibility to implement policies for fair and ethical business behavior of all forex -related institutions within the jurisdictions. In turn all forex brokers and signal sellers have to operate in strict compliance with the rules and standards laid by the forex regulators otherwise their activities is regarded unlawful. These regulatory bodies strive to formulate rules and laws modified and amended to fit the growing field of forex. [58] (Forex Regulators)

Especially, the spot forex has been a challenge to the regulators. For example, in 2000 congress the commodity modernization Act (CFMA) this amended the commodity exchange act (CEA). Through this amendment, the CFTC was able to regulate retail forex contracts that are either futures or options Another rule passed by the congress in July 15, 2010 is the Dodd- Frank Wall Street Reform and consumer protection act, H.R 4173 2010. In this rule those who are trading in forex are changed with responsibility marking sure they register, keep records, and report their trading activities. As Mentioned in the Investment Law Group this rule defined a new category of registrant called the Retail Foreign Exchange Dealer and Impose Restrictions of RFEDs as well as futures Commission Merchants (FCMS) introducing brokers (IBS) commodity pool operators CPOs, Commodity Trading Advisor (CTAs) and the Associated persons (Aps) all that those who are involved in over the counter forex market. [59] (Investment Law Group)

However, despite the regulatory Institutions, rules, regulation, policies and amendments, there is still a great risk of traders or investors to run into scam. So it is important that all traders, especially newbies, look for brokers that have registered and are following the company's policies as required by the US Laws.

[57] (Maeda and Burrel)
[58] (Forex Regulators)
[59] (Investment Law Group). - http://www.investmentlawgroup.com/

4.4 Licensing Forex Trading

Forex trade business (broker) license is one of the most important criteria used to check whether a trading business authenticity and reliability. In fact running a forex trading business without a license can be regarded as regulatory violation. Each country and states have different criteria on how to obtain a license so a trader must go by the licensing regulations of the country and states they have set their forex trading business.

In the United States, traders must apply for the trading license through the National Futures Association (NFA), the Commodities Futures Trading Commission (CFTC) and The Security and Exchange Commission (SEC). Obtaining a forex trading license can be difficult and tedious because of the vast regulations created by the NFTC and SEC. To apply for the broker license, the trader must submit the memorandum and article of association, internal and external documents, the required examinations results and the business plan. A trader might consider hiring a legal assistance to advise them of the step of registration and licensing brokerage company they are opening.

There are several categories of licenses but relevant license that a forex broker needs is the series 3 license and series 30 license.

Series 3 License a test administered by the National commodity futures for a broker to get a job with forex business companies. Since part of the goal is to install a brokerage company, it is required to pass the exams in order satisfy the proficiency requirements. The series 3 test consists of 120 questions, the completion time is 150 minutes and it cost $115.[60] (Schramm)The exams are administered in two parts testing market knowledge and regulations that govern the business. The topics covered are the cash contracts, forward, and future, options, and swap contracts, it tests a broker understands strategies and how to calculate gains and loses. The test also covers the initial deposit margin and additional margins, the trading floor procedures and the functions of traders and brokers. The trader must be complaisant with the National Futures Association's rules and regulations. More the exam test how traders understand reporting rules and customer account.[61] (Hunter)

The series 3 licenses is a pre-requisite requirement for a broker, but opening an independent brokerage business, would require to have a series 30 licenses. A series 30 license is issued by the Financial Industry Regulation Authority to a qualified person to manage a branch office for financial company. To obtain a series 30, one would need to take a series 30 exams administered by the FINRA. The exams

[60] (Schramm)
[61] (Hunter)

include 50 questions, takes 60 minutes and we would need to get 70% of the test in order to get the series 30 license. The topics tested are the management knowledge, restrictions on marketing, disclosure rules, and general rules and regulations. [62] (What is Series 30?)

[62](What is Series 30?)

4.5 Marketing.

Once the business is set up, the next step is the marketing and advertising of products and services. It is a huge part of a business' success to attract customers by not only having a good product but also providing trust and comfort and an overall satisfying customer service experience. A company cannot rely only on word of mouth to keep new customers coming, it is important to use all modern and conventional ways available to advertise and promote the business.

What to Advertise

The main item that investors are looking for is the trading performance, but profits alone can't attract a good customer base. The goal is to advertise a good winning percentage that shows consistency without too much risk, a drawdown that is small and does not reflect too much loss of capital and complete client disclosure, meaning that all trades including past performance reports will be available with full transparency. It is also important to show proper implementation of a trading system that operates on strict trading rules, such as not having overnight trades and working on pre-determined hours. It is also crucial for the investor to understand the risks and volatility involved with trading FOREX, so although some months will be losses and other months will be gains, the idea is to maintain a system that yields profits on a yearly basis.

How to advertise the product

The first step is to have a marketing plan for advertising. This plan consists of a dedicated amount of time to promote the product, starting with the most cost efficient such as free social media site advertising, and ending in direct mail delivery and ad placement. The most cost efficient ideas can be repeated more often, while the most expensive ones can be repeated after longer periods of time, not only will this keep spending under budget, but it will allow for a more organized process.

There are many options for advertising products, including news articles, holding informative sessions, sponsoring small projects, distributing pamphlets, buying ad space in desired customer targeted websites, and social media sites. When advertising, it is important to keep in mind the target audience, and the most efficient advertising sources for reaching them. For example, as an investing company, buying ad space in business related magazines and websites can produce more customers than in sports related media.

Chapter 5: Conclusions & Recommendations

5.1 Conclusion

Without any exposure to the trading world, a currency pair chart initially seemed like random noise, however this project helped group members filter and extract information about the many layers of factors that are constantly moving the supply and demand of a currency; from countries declaring bankruptcy in Europe, to population demographics in Asia, a ripple effect is seen affecting the Market as a whole. Through this project the group gained a much stronger grasp of the world's economy and the way in which it directly influences the individual person.

The powerful Idea of simplifying large amounts of information was extremely useful, and it was a concept that stuck with the group throughout the development of the system. The goal to eliminate the need to look at individual charts to see the movement of currency pairs was fulfilled by the creation of the FXCM Multiple Currency Overview Indicator, allowing for one window to provide the trader with market history in 28 currency pairs up to 1 year previous from the current date.

The results from individual trading showed promise that a money management company can be started and research done in the legal structure, licensing, performance and marketing requirements of starting a company gave the group more knowledge of the steps necessary to take where they looking to start a business.

Exposure to the FOREX market and the analysis of its behavior has made the possibility of actual world investment a reality. By practicing with simulated trading accounts, the total net profit achieved was $ **12,532.8,** which provided huge motivation to the group and one of the members made the decision to open a live trading account with the purpose of generating partial income.

5.2 Recommendation for Future IQP

The FXCM Multiple Currency Overview has the capabilities of being a very versatile tool, and there are several additions and modifications to the code provided to make the indicator more customizable. A background in C or C++ is recommended in pursuing this project, as they have nearly the same syntax as MQL4.

Familiarization with MQL4

In order to become familiar with MQL4, as well as the behavior of the FXCM Multiple Currency Overview, it is recommended to create a separate copy of the FXCM Multiple Currency Overview. After making a separate copy, spend time modifying each parameter and observe the corresponding effects. Follow a similar process with each function within the code to gain understanding of what each line of code is accomplishing. Writing simple, small indicators separate to the main project also aids in understanding of the language and behavior of MQL4 in Metatrader.

Mixing and Matching Indicators

Currently, the trade order conditions on the FXCM Multiple Currency Overview are determined by a triple moving average indicator, using one SMA and two EMAs. One possible addition to this program is to add several indicators options, allowing the user to choose which indicator, or combination there-of, will determine the result of the trade order.

Improved Display Customization

The spacing and font size of the graphs can be set by the user, but all of the content is locked in place. In order to allow greater flexibility across different display sizes, as well as user interests, the program could be modified to implement more display options. These options include selecting which currencies are displayed, as well as which order they are displayed in.

Automated Trading Across Multiple Currencies

Due to the syntax of MQL4, this option may or may not be possible, and will require research into programming Expert Advisors in Metatrader. This modification would not only allow automated trading across multiple currencies, but would also allow back testing and optimization.

References

n.d. <fxstreet.com>.

Achelis, Steven B. "Technical Analysis from A to Z." McGraw Hill Professional, n.d. Web. <http://sce.uhcl.edu/boetticher/ML_DataMining/TechAnalysisAtoZ.PDF>.

Aroon Indicator. 2007-12. OnlineTradingConcepts.com. <http://www.onlinetradingconcepts.com/TechnicalAnalysis/Aroon.html>.

Blackstone, Brian. "Europe Bank Chief's Remarks Damp Euro." *The Wall Street Journal* (2013). Web. <http://online.wsj.com/news/articles/SB10001424127887324590904578289650217227008>.

Bordo, Michael D. "The Bretton Woods International Monetary System: A Historical." *National Bureau of Economic Research*. Vol. A Retrospective on the Bretton Woods System: Lessons for International Monetary Reform. 1993. 3-108. <http://www.nber.org/chapters/c6867.pdf>.

Bressert, Walter. "The Art of Surviving Through Money Managemet." *Futures* (1981). Web. <http://www.walterbressert.com/pdf/MoneyManagement.PDF>.

ChartSchool - . "See Specific Citation for webpage name." 1999-2013. *StockCharts.com - ChartSchool.* 2013. <http://stockcharts.com/school/doku.php?id=chart_school>.

ChartSchool. *Moving Averages - Simple and Exponential*. 1999-2013. <http://stockcharts.com/school/doku.php?id=chart_school:technical_indicators:moving_averages>.

Christopher, Neely. "Technical Analysis in the foreign exchange market: a layman's guide." *Federal Reserve Bank of St Louise* (2008): 23-38.

Cody, Tom, Dem A. Hopkins and Lawrence A. Perlman. *Guide To Limited Liability Companies*. Chicago: CCH Group, 2007.

"Corporation Types." Companies Incorporated, 2013. Web. <http://www.companiesinc.com/corporation/types.asp>.

"Definition -." 2013. Investopedia.com Dictionary. <http://www.investopedia.com/dictionary/>.

"Economic Calendar." 2013. FXstreet.com. The Forex Market. <http://www.fxstreet.com/economic-calendar/>.

Everett, John O., Cherie Hennig and Nancy Nichols. *Contemporay tax practice: Research, Planning and Strategies*. CCH, 2008.

Fabozzi, Frank J. *The Basics of Finance + Web Site*. John Wiley & Sons, 2010. Web. <http://www.bazneshaste.com/upload/modules/articles/pdfs/7.pdf>.

Forex Factory. 2013. Web-Calendar. <http://www.forexfactory.com/calendar.php>.

"Forex Fundamental Analysis." Forex Metrics, 2005-13. Web. <http://www.forexmetrics.com/forex-fundamental-analysis.html#top>.

"Forex Market Hours." forexchartspro.com, 2004-2013. web. <http://www.forexchartspro.com/learn/forex-basics/forex-market-hours>.

"Forex Regulators." Globe Gain, 2011-13. Web. <http://globegain.com/brokers/forex-regulators>.

"Forex Taxes." WLF Network, n.d. Web. <http://www.futurestrading.com.br/forex_taxes.htm>.

ForexOptions. "Taxation of Foreign Currency Trading Demystified." n.d. Web. <http://www.fxop.com/Forex%20Taxation.html>.

forextraders.com. "The Best and the Worst Times to Trade Forex." Blue Stripe Incubator, 13 Jun. 2013. Web. <http://www.forextraders.com/forex-strategy/the-best-and-worst-times-to-trade-forex>.

Forrester, Jim. "Trade as a Business Entity and Reap the Benefits." *TradersLog* (2005-10). <http://www.traderslog.com/business/>.

Hatten, Timothy S. *Small Business Management: Entrepreneurship and Beyond* . Ed. Acuna. Melissa. South-Western, 2012.

Hinson, Chris. "The Difference Between a Partnership and a Limited Partnership." ExpertHub.com, 2013. Web Article. <http://www.alllaw.com/articles/business_and_corporate/article12.asp>.

Hunter, Nick. "Series 3 Commodities Futures Exam." n.d. Web. <http://www.brokerjobs.com/s3.htm>.

Investopedia US. "Money Market: What Is It?" *Money Market*. ValueClick, 2013. investopedia.com-Personal Finance. <http://www.investopedia.com/university/moneymarket/moneymarket1.asp>.

IRS-Partnerships. 27 Jun. 2013. Web. 2013. <http://www.irs.gov/Businesses/Small-Businesses-&-Self-Employed/Partnerships>.

"Ishaq Dar." n.d. <http://en.wikipedia.org/wiki/Ishaq_Dar>.

Jack, Schannep. *Dow Theory for the 21st Century: Technical Indicators for Improving Your Investment Result*. Hoboken NewJersey: John Wiley and Sons,Inc, 2008.

Jason Hartman Team. "Why a “Gold Standard” is Not the Answer." n.d. *The Complete Solution for Real Estate Investors.* Online Blog. 12 Feb 2013. <http://www.jasonhartman.com/why-a-gold-standard-is-not-the-answer-2/>.

Jeff, Madura. *Financial Market and Institutions*. 10 Edition. Mason: South Western Cengage Learning, 2012.

John, Murphy. *Technical Analysis of the Financial Market: A comprehensive Guide*. New York: Penquin , 1999.

Krivo, Richard. "Quieting Market Noise." DailyFX, 21 May 2011. <http://www.dailyfx.com/forex/education/trading_tips/post_of_the_day/2011/05/31/Quieting_Market_Noise.html>.

—. "Quieting Market Noise." DailyFX, 31 May 2011. <http://www.dailyfx.com/forex/education/trading_tips/post_of_the_day/2011/05/31/Quieting_Market_Noise.html>.

Kubasek, Nancy, Bartley A. Brennan and M.Neil Browne. *The Legal Environment of Business: A Critical Thinking Approach*. Pearson Education, 2012.

Lakhani, Jay. "Stochastic Oscillator." n.d. Web. <http://www.4xmentor.net/school/Stochastic%20Oscillator.pdf>.

Langager, Chad and Casey Murphy. "Dow Theory: Introduction." Investopedia US, 2013. Web. <http://www.investopedia.com/university/dowtheory/>.

Lessambo, Felix. *The International Banking System; Capital Adequacy, Core Businesses and Risk Management*. Macmillan, 2013.

"Limited Partnership." *QuickMBA- Law and Business*. Internet Center for Management and Business Admin., 1999-2010. Web- QuickMBA. <http://www.quickmba.com/law/partnership/limited/>.

Maeda, Martha and Jamaine Burrel. *The complete Guide to Currency Trading and Investing: How to Earn High Rates of Returns Safely and Take Control of Your Financial Investment* . Atlantic Publishing Group , 2011.

Milton, Adam. *TRIX*. n.d. Web. 2013. <http://daytrading.about.com/od/indicators/a/Trix.htm>.

"Momentum." Vizhon, n.d. Web- IncredibleCharts. <http://www.incrediblecharts.com/indicators/momentum.php>.

Morah, Chizoba. "Why are most bonds traded on the secondary market "over the counter"?" ValueClick, 29 Jan. 2010. Online Posting to Investopedia FAQ-Bonds. <http://www.investopedia.com/ask/answers/09/bond-over-the-counter.asp#ixzz2NOCZvgUu>.

Moving Average Envelopes. 2007-2012. OnlineTradingConcepts.com. 2013. <http://www.onlinetradingconcepts.com/TechnicalAnalysis/MovingAverageEnvelopes.html>.

Nadig, Dave, Matt Hougan and Lara Crigger. "Currency: The Overlooked Asset Class." *Index Universe* (2009). Web. <http://www.indexuniverse.com/publications/journalofindexes/joi-articles/6745-currency-the-overlooked-asset-class.html>.

O'Harrow, Robert. "A Primer on Financial Derivatives." *The Washington Post* (2010). Web. <http://www.washingtonpost.com/wp-dyn/content/article/2010/04/20/AR2010042004666.html>.

Paul, Neely J Christopher and Weller A. *Technical Analysis in the Foreign Exchange Market*. St Louis, 2 May 2011.

Piper, Mike. *Surprisingly Simple: Independent contractor, Sole Proprietor abd LLC Taxes Explained in 100 papes or Less*. Illinoies Chicago: Simple Subject LLC, 2010.

Radzicki, Michael J. "Tpic 5 & 6: Trading System Development- Overview & Details." Worcester Polytechnic Institute, 2012. Web-PPT.

Rene, Stulz M. "Third quorter 2005." n.d. <http://fisher.osu.edu/fin/faculty/stulz/.../milkeninstitute_pubpaper.pdf>.

Sabodin, Alexander. "The Alligator Indicator." *Technical Analysis of Stocks & Commodities* (2008). Web. <http://www.traders.com/Documentation/FEEDbk_docs/2008/01/ForexFocus/FOREXfocus.html>.

Schenk, Deborah H. *Federal Taxation of S corporations*. New York: ALM Property, 2005.

Schramm, Melinda H. *The Complete IB Handbook*. CME Group, MHS Capital Resources, 2012. Web PDF. <http://www.rgbrokerservices.com/assets/IBhandbook.pdf>.

"Session Overlaps." babypips.com, 2013. web. <http://www.babypips.com/school/preschool/when-can-you-trade-forex/session-overlaps.html>.

"Stochastics Indicator." ForexIndicators, n.d. Web. <http://www.forexindicators.net/stochastics-indicator.html>.

Thorp, Wayne A. "ID'ing When to Buy and Sell Using the Stochastic Oscillator." *AAII Journal* (2000). Web. <http://cdn.preterhuman.net/texts/unsorted2/Stock%20books%20058/Wayne%20A.%20Thorp/Wayne%20A.%20Thorp%20-%20Technical%20Analysis.pdf>.

TradeKing Group. "Understanding the Tax Implications of Trading." TradeKing Group, 2013. Web. <https://www.tradeking.com/education/trading/tax-implications-of-trading>.

"Trading Sessions." 2013. web-babypips.com. <http://www.babypips.com/school/preschool/when-can-you-trade-forex/trading-sessions.html#ixzz2OxpXydPh>.

What is Series 30? 2003-13. Conjecture Corporation. Web. 2013. <http://www.wisegeek.net/what-is-series-30.htm>.

Wright, Charlie F. *Trading as a Business*. 1998. Web.

Yahoo News. 2013. Web. <http://finance.yahoo.com/news/category-currencies>.

Attachments

Attachment I: Individual Trading Records

Mariela Qirici

Forex Capital Markets, LLC

Account: 70660309 **Name: Mariela Qirici** **Currency: EUR** **Leverage: 1:50** **2013 October 28, 15:01**

Closed Transactions:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Close Time	Price	Commission	Taxes	Swap	Profit
33854661	2013.10.23 22:10	balance	Deposit										100 000.00
33856300	2013.10.24 00:14	buy	1.00	audusd	0.96392	0.00000	0.00000	2013.10.24 02:07	0.96547	0.00	0.00	0.00	112.44
33856304	2013.10.24 00:14	buy	1.00	eurchf	1.23006	0.00000	0.00000	2013.10.24 11:16	1.22999	0.00	0.00	0.00	-5.69
33856325	2013.10.24 00:15	buy	1.00	audnzd	1.14593	0.00000	0.00000	2013.10.24 04:21	1.14658	0.00	0.00	0.00	39.68
33856473	2013.10.24 00:37	buy	1.00	gbpnzd	1.92167	0.00000	0.00000	2013.10.24 03:55	1.92285	0.00	0.00	0.00	72.07
33856475	2013.10.24 00:38	buy	1.00	eurnzd	1.63874	0.00000	0.00000	2013.10.24 07:08	1.64040	0.00	0.00	0.00	101.19
33857969	2013.10.24 02:07	buy	1.00	cadjpy	93.869	0.000	0.000	2013.10.24 02:18	93.930	0.00	0.00	0.00	45.40
33857970	2013.10.24 02:08	buy	1.00	cadchf	0.85987	0.00000	0.85619	2013.10.24 21:15	0.85619	0.00	0.00	0.00	-298.87
33857974	2013.10.24 02:08	buy	1.00	nzdjpy	82.160	0.000	0.000	2013.10.24 02:18	82.213	0.00	0.00	0.00	39.45
33857976	2013.10.24 02:09	buy	1.00	nzdchf	0.75238	0.00000	0.00000	2013.10.24 02:19	0.75285	0.00	0.00	0.00	38.22
33858012	2013.10.24 02:12	buy	1.00	audusd	0.96582	0.00000	0.00000	2013.10.24 02:18	0.96636	0.00	0.00	0.00	39.16
33858020	2013.10.24 02:13	buy	1.00	eurjpy	134.322	0.000	0.000	2013.10.24 02:18	134.384	0.00	0.00	0.00	46.14
33858272	2013.10.24 02:23	buy	1.00	audjpy	94.185	0.000	0.000	2013.10.24 02:37	93.887	0.00	0.00	0.00	-222.07
33858307	2013.10.24 02:25	buy	1.00	usdjpy	97.452	0.000	0.000	2013.10.24 02:37	97.339	0.00	0.00	0.00	-84.21
33858449	2013.10.24 02:29	buy	1.00	gbpcad	1.68045	0.00000	0.00000	2013.10.24 07:41	1.68078	0.00	0.00	0.00	23.03
33858827	2013.10.24 02:38	buy	1.00	eurgbp	0.85156	0.00000	0.00000	2013.10.24 07:04	0.85229	0.00	0.00	0.00	85.65
33858888	2013.10.24 02:39	buy	1.00	euraud	1.42932	0.00000	0.00000	2013.10.24 07:09	1.42976	0.00	0.00	0.00	30.77
33859443	2013.10.24 02:56	buy	1.00	usdjpy	97.338	0.000	0.000	2013.10.24 04:36	97.357	0.00	0.00	0.00	14.16
33859942	2013.10.24 03:06	buy	1.00	eurusd	1.37925	0.00000	0.00000	2013.10.24 07:04	1.38035	0.00	0.00	0.00	79.69
33861757	2013.10.24 07:30	buy	1.00	eurusd	1.37840	0.00000	0.00000	2013.10.24 10:18	1.38127	0.00	0.00	0.00	207.78
33861955	2013.10.24 03:50	sell limit	1.00	usdjpy	97.427	0.000	0.000	2013.10.24 05:29	97.427	deleted [hedge is prohibited]			
33863682	2013.10.24 04:23	buy	1.00	audnzd	1.14731	0.00000	0.00000	2013.10.24 07:18	1.14760	0.00	0.00	0.00	17.67
33864211	2013.10.24 04:37	buy	1.00	usdjpy	97.369	0.000	0.000	2013.10.24 07:04	97.379	0.00	0.00	0.00	7.44
33872487	2013.10.24 07:09	buy	1.00	gbpcad	1.68028	0.00000	0.00000	2013.10.24 07:41	1.68089	0.00	0.00	0.00	42.59
33872604	2013.10.24 07:10	buy	1.00	eurusd	1.38103	0.00000	0.00000	2013.10.24 10:17	1.38140	0.00	0.00	0.00	26.78
33873342	2013.10.24 07:21	buy	1.00	eurcad	1.43253	0.00000	0.00000	2013.10.24 08:38	1.43296	0.00	0.00	0.00	30.01
33873620	2013.10.24 07:24	buy	1.00	euraud	1.43071	0.00000	0.00000	2013.10.24 07:39	1.43125	0.00	0.00	0.00	37.73
33873979	2013.10.24 07:27	buy	1.00	audnzd	1.14844	0.00000	0.00000	2013.10.24 08:07	1.14896	0.00	0.00	0.00	31.61

33874315	2013.10.24 07:29	buy	1.00	usdchf	0.89209	0.00000	0.00000	2013.10.24 08:09	0.89267	0.00	0.00	0.00	47.20
33874530	2013.10.24 07:30	buy	1.00	eurjpy	134.226	0.000	0.000	2013.10.24 07:33	134.319	0.00	0.00	0.00	69.24
33875675	2013.10.24 07:36	buy	1.00	audusd	0.96377	0.00000	0.96205	2013.10.25 04:11	0.96205	0.00	0.00	0.00	-124.51
33880516	2013.10.24 08:18	buy	1.00	nzdusd	0.83743	0.00000	0.83075	2013.10.28 08:03	0.83076	0.00	0.00	0.00	-482.99
33880638	2013.10.24 08:20	buy	1.00	gbpusd	1.61848	0.00000	0.00000	2013.10.24 08:37	1.61925	0.00	0.00	0.00	55.85
33881921	2013.10.24 08:37	buy	1.00	gbpusd	1.61940	0.00000	0.00000	2013.10.24 16:31	1.62008	0.00	0.00	0.00	49.24
33882152	2013.10.24 08:41	buy	1.00	eurchf	1.22922	0.00000	0.00000	2013.10.24 11:16	1.23000	0.00	0.00	0.00	63.41
33882165	2013.10.24 08:41	buy	1.00	audjpy	93.759	0.000	0.000	2013.10.24 16:20	93.504	0.00	0.00	0.00	-189.72
33882215	2013.10.24 08:42	buy	1.00	nzdusd	0.83747	0.00000	0.83075	2013.10.28 08:03	0.83076	0.00	0.00	0.00	-485.86
33882249	2013.10.24 08:43	buy	1.00	usdjpy	97.395	0.000	0.000	2013.10.24 12:47	97.436	0.00	0.00	0.00	30.52
33889322	2013.10.24 10:24	buy	1.00	eurjpy	134.440	0.000	134.415	2013.10.25 06:00	134.195	0.00	0.00	0.00	-182.57
33889964	2013.10.24 10:32	buy	1.00	cadjpy	93.578	0.000	93.465	2013.10.27 21:03	93.467	0.00	0.00	0.00	-82.27
33898078	2013.10.24 12:24	buy	1.00	audusd	0.96249	0.00000	0.96205	2013.10.25 04:11	0.96205	0.00	0.00	0.00	-31.86
33900887	2013.10.24 12:49	buy	1.00	usdjpy	97.439	0.000	0.000	2013.10.24 16:27	97.362	0.00	0.00	0.00	-57.28
33902305	2013.10.24 13:02	buy	1.00	eurusd	1.38007	0.00000	0.00000	2013.10.24 13:05	1.38176	0.00	0.00	0.00	122.31
33903504	2013.10.24 13:09	buy	1.00	audjpy	93.365	0.000	0.000	2013.10.24 16:20	93.504	0.00	0.00	0.00	103.42
33903551	2013.10.24 13:09	buy	1.00	usdjpy	97.296	0.000	0.000	2013.10.24 16:27	97.362	0.00	0.00	0.00	49.10
33903579	2013.10.24 13:09	buy	1.00	cadjpy	93.501	0.000	93.465	2013.10.27 21:03	93.471	0.00	0.00	0.00	-22.24
33921857	2013.10.24 16:23	buy	1.00	gbpchf	1.44402	0.00000	0.00000	2013.10.24 16:48	1.44463	0.00	0.00	0.00	49.56
33921901	2013.10.24 16:25	buy	1.00	audnzd	1.15111	0.00000	0.00000	2013.10.24 17:10	1.15144	0.00	0.00	0.00	19.96
33922502	2013.10.24 16:50	buy	1.00	gbpchf	1.44473	0.00000	0.00000	2013.10.24 16:52	1.44487	0.00	0.00	0.00	11.37
33923739	2013.10.24 17:13	buy	1.00	audchf	0.85777	0.00000	0.00000	2013.10.24 17:27	0.85834	0.00	0.00	0.00	46.29
33923768	2013.10.24 17:14	buy	1.00	nzdcad	0.87093	0.00000	0.87035	2013.10.28 18:45	0.86745	0.00	0.00	0.00	-241.40
33925864	2013.10.24 17:35	buy	1.00	audchf	0.85845	0.00000	0.85585	2013.10.25 09:27	0.85585	0.00	0.00	0.00	-210.85
33935061	2013.10.24 19:05	buy	1.00	chfjpy	109.129	0.000	109.040	2013.10.25 15:27	109.043	0.00	0.00	0.00	-63.96
33935076	2013.10.24 19:07	buy	1.00	eurpln	4.18311	0.00000	4.18490	2013.10.25 06:33	4.18490	0.00	0.00	0.00	42.77
33992705	2013.10.24 21:22	balance	Rollover - 33875675										2.90
33992706	2013.10.24 21:22	balance	Rollover - 33898078										2.90
33992707	2013.10.24 21:22	balance	Rollover - 33882215										2.80
33992708	2013.10.24 21:22	balance	Rollover - 33880516										2.80
33992709	2013.10.24 21:22	balance	Rollover - 33889322										0.20
33992712	2013.10.24 21:22	balance	Rollover - 33935061										-0.30
33992713	2013.10.24 21:22	balance	Rollover - 33889964										1.20
33992714	2013.10.24 21:22	balance	Rollover - 33903579										1.20
33992715	2013.10.24 21:22	balance	Rollover - 33925864										3.30
33992716	2013.10.24 21:22	balance	Rollover - 33935076										-11.20
33992717	2013.10.24 21:22	balance	Rollover - 33857970										1.40
33992718	2013.10.24 21:22	balance	Rollover - 33923768										1.80
34042403	2013.10.25 04:03	buy	1.00	eurusd	1.38143	0.00000	1.38175	2013.10.25 04:19	1.38177	0.00	0.00	0.00	24.61
34042485	2013.10.25 04:04	buy	1.00	gbpusd	1.62398	0.00000	1.62440	2013.10.25 04:34	1.62441	0.00	0.00	0.00	31.09
34042550	2013.10.25 04:04	buy	1.00	usdjpy	97.214	0.000	97.300	2013.10.25 11:16	97.302	0.00	0.00	0.00	65.56
34042664	2013.10.25 04:06	buy	1.00	gbpcad	1.69043	0.00000	1.69070	2013.10.25 07:34	1.69070	0.00	0.00	0.00	18.73
34042725	2013.10.25 04:06	buy	1.00	eurcad	1.43857	0.00000	1.43888	2013.10.25 04:19	1.43888	0.00	0.00	0.00	21.54
34043765	2013.10.25 04:18	buy	1.00	eurjpy	134.118	0.000	134.205	2013.10.25 08:44	134.205	0.00	0.00	0.00	64.83
34044708	2013.10.25 04:30	buy	1.00	usdchf	0.89061	0.00000	0.89125	2013.10.25 07:04	0.89126	0.00	0.00	0.00	52.81
34050399	2013.10.25 05:51	buy	1.00	chfjpy	109.089	0.000	109.040	2013.10.25 15:27	109.043	0.00	0.00	0.00	-34.21
34050548	2013.10.25 05:53	buy	5.00	eurjpy	134.179	0.000	134.205	2013.10.25 08:44	134.205	0.00	0.00	0.00	96.87

34050603	2013.10.25 05:54	buy	5.00	eurjpy	134.165	0.000	134.205	2013.10.25 08:44	134.205	0.00	0.00	0.00	149.03
34052078	2013.10.25 06:08	buy	1.00	audchf	0.85515	0.00000	0.85585	2013.10.25 09:27	0.85593	0.00	0.00	0.00	63.26
34052317	2013.10.25 06:09	buy	1.00	audchf	0.85511	0.00000	0.85585	2013.10.25 09:27	0.85588	0.00	0.00	0.00	62.45
34052494	2013.10.25 06:11	buy	1.00	audchf	0.85494	0.00000	0.85585	2013.10.25 09:27	0.85590	0.00	0.00	0.00	77.85
34053370	2013.10.25 06:24	buy	5.00	usdcad	1.04250	0.00000	1.04265	2013.10.25 06:50	1.04267	0.00	0.00	0.00	59.01
34053949	2013.10.25 06:29	buy	5.00	chfjpy	108.980	0.000	109.040	2013.10.25 15:27	109.043	0.00	0.00	0.00	234.29
34055245	2013.10.25 06:47	buy	5.00	gbpcad	1.69049	0.00000	1.69070	2013.10.25 07:34	1.69072	0.00	0.00	0.00	79.80
34055416	2013.10.25 06:48	buy	5.00	gbpcad	1.69021	0.00000	1.69070	2013.10.25 07:34	1.69074	0.00	0.00	0.00	183.89
34078345	2013.10.25 11:34	buy	1.00	eurusd	1.37985	0.00000	1.38120	2013.10.25 19:28	1.38120	0.00	0.00	0.00	97.74
34154114	2013.10.28 03:09	buy	1.00	euraud	1.43710	0.00000	1.43905	2013.10.28 12:59	1.43905	0.00	0.00	0.00	135.51
34154152	2013.10.28 03:09	buy	1.00	gbpnzd	1.94830	0.00000	1.95070	2013.10.28 06:56	1.94869	0.00	0.00	0.00	23.47
34154208	2013.10.28 03:10	buy	1.00	nzdchf	0.74214	0.00000	0.74300	2013.10.28 09:18	0.74301	0.00	0.00	0.00	70.49
34156891	2013.10.28 03:46	buy	1.00	audnzd	1.15660	0.00000	1.15633	2013.10.28 03:47	1.15634	0.00	0.00	0.00	-15.64
34184809	2013.10.28 09:47	buy	1.00	eurusd	1.38125	0.00000	0.00000	2013.10.28 18:45	1.38065	0.00	0.00	0.00	-43.46
34197529	2013.10.28 13:19	buy	1.00	gbpusd	1.61595	0.00000	0.00000	2013.10.28 18:54	1.61576	0.00	0.00	0.00	-13.77
34197799	2013.10.28 13:28	buy	1.00	eurcad	1.43967	0.00000	0.00000	2013.10.28 14:07	1.44075	0.00	0.00	0.00	74.96
34197801	2013.10.28 13:28	buy	1.00	gbpnzd	1.94922	0.00000	0.00000	2013.10.28 13:39	1.95007	0.00	0.00	0.00	51.06
34197814	2013.10.28 13:29	buy	1.00	eurnzd	1.66318	0.00000	1.66440	2013.10.28 13:35	1.66440	0.00	0.00	0.00	73.30
34198373	2013.10.28 13:41	buy	1.00	eurnzd	1.66526	0.00000	0.00000	2013.10.28 13:54	1.66527	0.00	0.00	0.00	0.60
34198457	2013.10.28 13:43	buy	1.00	nzdcad	0.86484	0.00000	0.87035	2013.10.28 18:45	0.86745	0.00	0.00	0.00	181.05
34198462	2013.10.28 13:43	buy	1.00	eurusd	1.37793	0.00000	0.00000	2013.10.28 18:46	1.38066	0.00	0.00	0.00	197.73
34198509	2013.10.28 13:44	buy	1.00	usdjpy	97.785	0.000	0.000	2013.10.28 18:57	97.643	0.00	0.00	0.00	-105.32
34198595	2013.10.28 13:45	buy	1.00	eurjpy	134.737	0.000	0.000	2013.10.28 18:46	134.781	0.00	0.00	0.00	32.65
34198910	2013.10.28 13:53	buy	1.00	nzdcad	0.86482	0.00000	0.87035	2013.10.28 18:46	0.86741	0.00	0.00	0.00	179.65
										0.00	0.00	0.00	1 333.98
												Closed P/L:	**1 333.98**

Open Trades:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P		Price	Commission	Taxes	Swap	Profit
34197809	2013.10.28 13:29	buy	1.00	audcad	1.00057	0.00000	1.00255		1.00022	0.00	0.00	0.00	-24.28
34198652	2013.10.28 13:47	buy	1.00	audjpy	93.546	0.000	0.000		93.542	0.00	0.00	0.00	-2.97
34197567	2013.10.28 13:20	buy	1.00	audusd	0.95822	0.00000	0.00000		0.95789	0.00	0.00	0.00	-23.91
34216098	2013.10.28 18:59	buy	1.00	eurjpy	134.859	0.000	0.000		134.809	0.00	0.00	0.00	-37.09
34216040	2013.10.28 18:58	buy	1.00	gbpusd	1.61594	0.00000	0.00000		1.61566	0.00	0.00	0.00	-20.28
34198625	2013.10.28 13:46	buy	1.00	usdcad	1.04488	0.00000	0.00000		1.04407	0.00	0.00	0.00	-56.20
										0.00	0.00	0.00	-164.73
												Floating P/L:	**-164.73**

Working Orders:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Market Price	

No transactions

Sebastian Franco

Forex Capital Markets, LLC

Account: 50135087 **Name: sfranco** **Currency: USD** **Leverage: 1:50** **2013 October 25, 01:19**

Closed Transactions:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Close Time	Price	Commission	Taxes	Swap	Profit
23225855	2013.10.10 20:54	balance	Deposit										100 000.00
23226059	2013.10.10 21:15	buy	1.00	eurjpy	132.810	0.000	0.000	2013.10.10 21:45	132.883	0.00	0.00	0.00	74.28
23286089	2013.10.11 03:26	buy	1.00	eurjpy	133.420	0.000	0.000	2013.10.11 20:07	133.474	0.00	0.00	0.00	54.82
23286248	2013.10.11 03:33	buy	1.00	nzdjpy	81.923	0.000	0.000	2013.10.11 20:06	81.991	0.00	0.00	0.00	69.03
23367123	2013.10.13 23:16	sell	1.00	euraud	1.43631	0.00000	0.00000	2013.10.14 02:11	1.43506	0.00	0.00	0.00	118.16
23373336	2013.10.14 03:00	buy	1.00	audusd	0.94591	0.00000	0.00000	2013.10.14 03:27	0.94657	0.00	0.00	0.00	66.00
23373678	2013.10.14 03:32	buy	0.01	audusd	0.94684	0.00000	0.00000	2013.10.14 04:16	0.94693	0.00	0.00	0.00	0.09
23374193	2013.10.14 04:22	sell	1.00	eurgbp	0.84847	0.00000	0.00000	2013.10.14 07:17	0.84889	0.00	0.00	0.00	-67.11
23376208	2013.10.14 07:19	sell	1.00	gbpusd	1.59763	0.00000	0.00000	2013.10.14 23:23	1.59686	0.00	0.00	0.00	77.00
23376376	2013.10.14 07:27	buy	1.00	eurchf	1.23572	0.00000	0.00000	2013.10.15 10:18	1.23694	0.00	0.00	0.00	133.16
23400461	2013.10.15 04:35	sell	1.00	usdcad	1.03333	0.00000	0.00000	2013.10.16 17:58	1.03309	0.00	0.00	0.00	23.23
23471082	2013.10.15 23:35	balance	Rollover - 23400461										1.20
23472874	2013.10.15 23:58	buy	10.00	audchf	0.87179	0.00000	0.00000	2013.10.16 18:52	0.87227	0.00	0.00	0.00	525.80
23474328	2013.10.16 00:50	buy	1.00	usdchf	0.91462	0.00000	0.00000	2013.10.16 01:35	0.91485	0.00	0.00	0.00	25.14
23480305	2013.10.16 07:40	buy	1.00	nzdcad	0.87437	0.00000	0.00000	2013.10.17 18:40	0.87489	0.00	0.00	0.00	50.54
23505609	2013.10.16 19:08	buy	1.00	gbpnzd	1.89240	0.00000	0.00000	2013.10.16 19:10	1.89252	0.00	0.00	0.00	10.11
23505987	2013.10.16 19:27	buy	1.00	gbpnzd	1.89390	0.00000	0.00000	2013.10.16 22:36	1.89456	0.00	0.00	0.00	55.56
23557101	2013.10.16 23:08	balance	Rollover - 23505987										-12.40
23557102	2013.10.16 23:08	balance	Rollover - 23480305										2.30
23568200	2013.10.17 07:26	buy	1.00	eurjpy	133.678	0.000	0.000	2013.10.17 07:29	133.689	0.00	0.00	0.00	11.19
23780613	2013.10.21 20:24	buy	5.00	usdjpy	98.219	0.000	0.000	2013.10.22 02:12	98.275	0.00	0.00	0.00	284.91
23780619	2013.10.21 20:25	buy	5.00	audjpy	94.859	0.000	0.000	2013.10.22 03:24	94.869	0.00	0.00	0.00	50.85

23780673	2013.10.21 20:29	buy	5.00	eurjpy	134.369	0.000	0.000	2013.10.22 02:16	134.399	0.00	0.00	0.00	152.54
23780887	2013.10.21 20:53	buy	10.00	audjpy	94.814	0.000	0.000	2013.10.22 03:25	94.867	0.00	0.00	0.00	539.05
23817787	2013.10.22 21:48	sell	1.00	gbpjpy	159.190	0.000	0.000	2013.10.23 06:55	157.581	0.00	0.00	0.00	1 651.51
23817849	2013.10.22 21:52	buy	1.00	eurgbp	0.84911	0.00000	0.00000	2013.10.23 06:56	0.85133	0.00	0.00	0.00	359.05
23817957	2013.10.22 22:02	buy	1.00	usdcad	1.02918	0.00000	0.00000	2013.10.23 06:56	1.03218	0.00	0.00	0.00	290.65
23817960	2013.10.22 22:03	buy	1.00	eurcad	1.41821	0.00000	0.00000	2013.10.23 06:56	1.42153	0.00	0.00	0.00	321.65
23991246	2013.10.23 22:54	balance	Rollover - 23817851										3.50
23991247	2013.10.23 22:54	balance	Rollover - 23817742										1.00
23999970	2013.10.24 01:31	sell	1.00	eurgbp	0.85184	0.00000	0.00000	2013.10.24 03:50	0.85155	0.00	0.00	0.00	46.98
24003077	2013.10.24 03:57	sell	1.00	usdjpy	97.283	0.000	0.000	2013.10.25 05:17	97.051	0.00	0.00	0.00	239.05
24016065	2013.10.24 12:30	buy	1.00	eurusd	1.38143	0.00000	0.00000	2013.10.25 05:17	1.38182	0.00	0.00	0.00	39.00
24016087	2013.10.24 12:31	buy	1.00	eurusd	1.38123	0.00000	0.00000	2013.10.25 05:18	1.38173	0.00	0.00	0.00	50.00
24073969	2013.10.24 23:03	balance	Rollover - 24016065										-1.00
24073970	2013.10.24 23:03	balance	Rollover - 24016087										-1.00
24073971	2013.10.24 23:03	balance	Rollover - 24003077										-0.80
24073974	2013.10.24 23:03	balance	Rollover - 23817851										3.50
										0.00	0.00	0.00	5 252.24
										Closed P/L:			**5 252.24**

Open Trades:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P		Price	Commission	Taxes	Swap	Profit
No transactions													
										0.00	0.00	0.00	0.00
										Floating P/L:			**0.00**

Working Orders:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Market Price	
No transactions									

Johnathan Baiden

Forex Capital Markets, LLC

Account: 70655043 **Name: jonathan baiden** **Currency: EUR** **Leverage: 1:50** **2013 October 28, 15:17**

Closed Transactions:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Close Time	Price	Commission	Taxes	Swap	Profit
32847798	2013.10.10 01:37	balance	Deposit										100 000.00
32847969	2013.10.10 01:53	buy	0.01	gbpusd	1.59432	0.00000	0.00000	2013.10.10 21:43	1.59704	0.00	0.00	0.00	2.01
32848657	2013.10.10 02:31	buy	0.01	gbpusd	1.59280	0.00000	0.00000	2013.10.10 21:43	1.59704	0.00	0.00	0.00	3.13
32876277	2013.10.10 15:06	sell	0.01	usdchf	0.90961	0.00000	0.00000	2013.10.10 21:42	0.91162	0.00	0.00	0.00	-1.63
32877273	2013.10.10 15:15	buy limit	0.01	usdchf	0.91010	0.91008	0.91038	2013.10.10 15:25	0.91010	deleted [hedge is prohibited]			
32893512	2013.10.10 23:06	buy	0.01	eurjpy	132.751	0.000	0.000	2013.10.11 00:23	132.966	0.00	0.00	0.00	1.62
32894492	2013.10.11 00:17	buy	0.01	eurjpy	132.932	0.000	0.000	2013.10.11 00:24	132.965	0.00	0.00	0.00	0.25
32894570	2013.10.11 00:30	buy	0.01	eurjpy	132.985	0.000	0.000	2013.10.11 01:40	133.118	0.00	0.00	0.00	1.00
32894680	2013.10.11 00:53	buy	0.01	gbpjpy	157.007	0.000	0.000	2013.10.11 01:40	157.255	0.00	0.00	0.00	1.86
32897042	2013.10.11 03:23	buy	0.01	gbpusd	1.59846	0.00000	0.00000	2013.10.11 10:38	1.59701	0.00	0.00	0.00	-1.07
32897045	2013.10.11 03:24	buy	0.01	gbpjpy	157.521	0.000	0.000	2013.10.11 10:38	156.989	0.00	0.00	0.00	-3.99
32897051	2013.10.11 03:25	buy	0.01	eurjpy	133.423	0.000	0.000	2013.10.11 10:38	133.440	0.00	0.00	0.00	0.13
32897064	2013.10.11 03:29	buy	0.01	eurgbp	0.84716	0.00000	0.00000	2013.10.11 09:57	0.84903	0.00	0.00	0.00	2.20
32897085	2013.10.11 03:31	buy	0.01	nzdjpy	81.894	0.000	0.000	2013.10.11 10:07	82.044	0.00	0.00	0.00	1.13
32897118	2013.10.11 03:36	buy	0.01	eurcad	1.40784	0.00000	0.00000	2013.10.11 09:58	1.41039	0.00	0.00	0.00	1.81
32949260	2013.10.13 11:57	balance	Balance correction										0.01
32953134	2013.10.13 21:58	buy	0.01	audjpy	92.610	0.000	0.000	2013.10.14 00:24	92.870	0.00	0.00	0.00	1.95
32953199	2013.10.13 22:00	buy	0.01	usdjpy	98.277	0.000	0.000	2013.10.14 00:24	98.305	0.00	0.00	0.00	0.21
32953236	2013.10.13 22:01	buy	0.01	gbpusd	1.59692	0.00000	0.00000	2013.10.14 00:24	1.59660	0.00	0.00	0.00	-0.24
32955306	2013.10.14 01:43	buy	0.01	gbpusd	1.59766	0.00000	0.00000	2013.10.14 07:16	1.59766	0.00	0.00	0.00	0.00
32955427	2013.10.14 01:54	buy	1.00	gbpjpy	157.063	0.000	0.000	2013.10.14 13:57	157.136	0.00	0.00	0.00	54.74
32955706	2013.10.14 02:26	buy	1.00	nzdusd	0.83279	0.00000	0.00000	2013.10.14 06:59	0.83420	0.00	0.00	0.00	103.95
32956251	2013.10.14 03:07	buy	1.00	audusd	0.94608	0.00000	0.00000	2013.10.14 07:03	0.94624	0.00	0.00	0.00	11.80
32966079	2013.10.14 12:52	buy	0.01	eurjpy	133.403	0.000	0.000	2013.10.14 16:37	133.488	0.00	0.00	0.00	0.64
32983065	2013.10.14 20:33	buy	0.01	usdcad	1.03502	0.00000	0.00000	2013.10.15 09:48	1.03561	0.00	0.00	0.00	0.42
33026497	2013.10.14 21:20	balance	Rollover - 32983065										-0.02
33030161	2013.10.14 21:58	buy	1.00	eurcad	1.40414	0.00000	0.00000	2013.10.16 11:34	1.40573	0.00	0.00	0.00	113.11
33030274	2013.10.14 22:00	buy	1.00	audcad	0.98274	0.00000	0.00000	2013.10.15 00:01	0.98429	0.00	0.00	0.00	110.46
33030434	2013.10.14 22:07	buy	1.00	usdcad	1.03576	0.00000	0.00000	2013.10.15 15:39	1.03724	0.00	0.00	0.00	105.74
33035447	2013.10.15 02:18	buy	1.00	usdjpy	98.531	0.000	0.000	2013.10.15 09:47	98.639	0.00	0.00	0.00	81.06
33035465	2013.10.15 02:20	buy	1.00	gbpjpy	157.477	0.000	0.000	2013.10.16 00:05	157.480	0.00	0.00	0.00	2.25
33072609	2013.10.15 19:24	buy	1.00	usdjpy	98.409	0.000	0.000	2013.10.16 00:05	98.536	0.00	0.00	0.00	95.34
33123489	2013.10.15 21:21	balance	Rollover - 33072609										0.10
33123490	2013.10.15 21:21	balance	Rollover - 33035465										0.80
33123491	2013.10.15 21:21	balance	Rollover - 33030161										-3.60

33133974	2013.10.16 00:14	buy	1.00	eurchf	1.23587	0.00000	0.00000	2013.10.16 14:04	1.23609	0.00	0.00	0.00	17.80
33147533	2013.10.16 06:59	buy	1.00	gbpusd	1.59900	0.00000	0.00000	2013.10.16 07:14	1.59968	0.00	0.00	0.00	50.30
33148975	2013.10.16 07:29	buy	1.00	euraud	1.41736	0.00000	0.00000	2013.10.16 09:57	1.41861	0.00	0.00	0.00	88.11
33149137	2013.10.16 07:32	buy	1.00	eurgbp	0.84532	0.00000	0.00000	2013.10.16 12:22	0.84613	0.00	0.00	0.00	95.73
33170805	2013.10.16 12:07	buy	1.00	usdchf	0.91048	0.00000	0.00000	2013.10.16 13:47	0.91204	0.00	0.00	0.00	126.28
33171700	2013.10.16 12:22	buy	1.00	cadjpy	94.900	0.000	0.000	2013.10.16 13:17	95.062	0.00	0.00	0.00	121.40
33171721	2013.10.16 12:23	buy	1.00	chfjpy	108.148	0.000	0.000	2013.10.16 14:24	108.054	0.00	0.00	0.00	-70.39
33171745	2013.10.16 12:24	buy	1.00	audnzd	1.13378	0.00000	0.00000	2013.10.16 21:26	1.13433	0.00	0.00	0.00	34.26
33171770	2013.10.16 12:25	buy	1.00	gbpchf	1.45916	0.00000	0.00000	2013.10.16 13:48	1.46052	0.00	0.00	0.00	110.09
33171783	2013.10.16 12:25	buy	1.00	eurgbp	0.84615	0.00000	0.00000	2013.10.16 14:08	0.84662	0.00	0.00	0.00	55.51
33171806	2013.10.16 12:26	buy	1.00	audjpy	93.868	0.000	0.000	2013.10.16 13:52	94.061	0.00	0.00	0.00	144.46
33171821	2013.10.16 12:26	buy	1.00	eurchf	1.23456	0.00000	0.00000	2013.10.16 14:04	1.23609	0.00	0.00	0.00	123.78
33180097	2013.10.16 14:11	buy	1.00	eurgbp	0.84704	0.00000	0.00000	2013.10.16 15:47	0.84737	0.00	0.00	0.00	38.94
33180971	2013.10.16 14:21	buy	1.00	euraud	1.41873	0.00000	0.00000	2013.10.17 02:55	1.42107	0.00	0.00	0.00	164.66
33181249	2013.10.16 14:25	buy	1.00	eurusd	1.35200	0.00000	0.00000	2013.10.16 18:15	1.35322	0.00	0.00	0.00	90.16
33181375	2013.10.16 14:28	buy	1.00	gbpchf	1.45993	0.00000	0.00000	2013.10.17 14:46	1.45731	0.00	0.00	0.00	-212.45
33181414	2013.10.16 14:29	buy	1.00	eurcad	1.40076	0.00000	0.00000	2013.10.17 07:59	1.40307	0.00	0.00	0.00	164.64
33181507	2013.10.16 14:30	buy	1.00	gbpcad	1.65434	0.00000	0.00000	2013.10.17 08:14	1.65505	0.00	0.00	0.00	50.59
33181536	2013.10.16 14:31	buy	1.00	gbpnzd	1.89571	0.00000	0.00000	2013.10.17 07:59	1.89931	0.00	0.00	0.00	223.43
33181609	2013.10.16 14:33	buy	1.00	eurpln	4.16696	0.00000	0.00000	2013.10.17 07:59	4.17160	0.00	0.00	0.00	111.23
33279217	2013.10.16 21:22	balance	Rollover - 33181375										3.60
33279218	2013.10.16 21:22	balance	Rollover - 33180971										-29.40
33279219	2013.10.16 21:22	balance	Rollover - 33181414										-9.30
33279220	2013.10.16 21:22	balance	Rollover - 33181507										-8.10
33279221	2013.10.16 21:22	balance	Rollover - 33181536										-30.90
33279222	2013.10.16 21:22	balance	Rollover - 33171745										-1.20
33279223	2013.10.16 21:22	balance	Rollover - 33181609										-27.30
33292227	2013.10.17 02:22	buy	1.00	gbpusd	1.59780	0.00000	0.00000	2013.10.17 05:56	1.59872	0.00	0.00	0.00	67.88
33292245	2013.10.17 02:23	buy	1.00	usdjpy	98.719	0.000	0.000	2013.10.17 08:00	97.901	0.00	0.00	0.00	-613.15
33294397	2013.10.17 03:37	buy	1.00	audjpy	94.195	0.000	0.000	2013.10.17 13:15	94.075	0.00	0.00	0.00	-89.80
33320094	2013.10.17 08:00	buy	1.00	eurusd	1.36358	0.00000	0.00000	2013.10.17 08:31	1.36246	0.00	0.00	0.00	-82.20
33320163	2013.10.17 08:00	buy	1.00	gbpusd	1.60674	0.00000	0.00000	2013.10.17 08:31	1.60846	0.00	0.00	0.00	126.23
33320605	2013.10.17 08:01	buy	1.00	nzdjpy	82.799	0.000	0.000	2013.10.17 10:04	83.033	0.00	0.00	0.00	175.25
33327342	2013.10.17 08:34	buy	1.00	usdchf	0.90557	0.00000	0.00000	2013.10.21 02:34	0.90283	0.00	0.00	0.00	-221.89
33327385	2013.10.17 08:34	buy	1.00	eurjpy	133.595	0.000	0.000	2013.10.17 12:42	133.680	0.00	0.00	0.00	63.58
33327486	2013.10.17 08:35	buy	1.00	usdcad	1.02988	0.00000	0.00000	2013.10.17 13:00	1.03128	0.00	0.00	0.00	99.47
33327514	2013.10.17 08:36	buy	1.00	eurchf	1.23366	0.00000	0.00000	2013.10.17 13:14	1.23318	0.00	0.00	0.00	-38.92
33327604	2013.10.17 08:36	buy	1.00	gbpchf	1.45543	0.00000	0.00000	2013.10.17 14:46	1.45729	0.00	0.00	0.00	150.82
33345000	2013.10.17 12:21	buy	1.00	audnzd	1.13332	0.00000	0.00000	2013.10.17 13:14	1.13335	0.00	0.00	0.00	1.86
33348787	2013.10.17 12:52	buy	1.00	gbpnzd	1.89978	0.00000	0.00000	2013.10.17 13:10	1.90062	0.00	0.00	0.00	52.20
33352280	2013.10.17 13:28	buy	1.00	usdcad	1.03089	0.00000	0.00000	2013.10.21 02:35	1.02936	0.00	0.00	0.00	-108.67
33352385	2013.10.17 13:29	buy	1.00	eurgbp	0.84709	0.00000	0.00000	2013.10.18 16:54	0.84674	0.00	0.00	0.00	-41.34
33352495	2013.10.17 13:30	buy	1.00	nzdusd	0.84936	0.00000	0.00000	2013.10.17 13:50	0.84987	0.00	0.00	0.00	37.33
33352687	2013.10.17 13:32	buy	1.00	cadjpy	94.921	0.000	0.000	2013.10.17 15:47	95.158	0.00	0.00	0.00	177.25
33369293	2013.10.17 15:21	buy	1.00	eurgbp	0.84575	0.00000	0.00000	2013.10.18 16:54	0.84674	0.00	0.00	0.00	116.92
33391531	2013.10.17 20:04	buy	1.00	usdjpy	97.942	0.000	0.000	2013.10.17 22:45	98.022	0.00	0.00	0.00	59.70
33391747	2013.10.17 20:05	buy	1.00	eurchf	1.23432	0.00000	0.00000	2013.10.18 08:51	1.23469	0.00	0.00	0.00	29.97

33391882	2013.10.17 20:06	buy	1.00	audjpy	94.313	0.000	0.000	2013.10.18 01:00	94.426	0.00	0.00	0.00	84.27
33391984	2013.10.17 20:07	buy	1.00	audnzd	1.13415	0.00000	0.00000	2013.10.17 20:41	1.13534	0.00	0.00	0.00	73.83
33441996	2013.10.17 21:21	balance	Rollover - 33391531										0.10
33441997	2013.10.17 21:21	balance	Rollover - 33327342										0.40
33441998	2013.10.17 21:21	balance	Rollover - 33391747										0.20
33441999	2013.10.17 21:21	balance	Rollover - 33352280										-2.20
33442000	2013.10.17 21:21	balance	Rollover - 33352385										-1.90
33442001	2013.10.17 21:21	balance	Rollover - 33369293										-1.90
33442002	2013.10.17 21:21	balance	Rollover - 33391882										3.20
33449528	2013.10.17 23:06	buy	1.00	nzdusd	0.84611	0.00000	0.00000	2013.10.18 07:47	0.84727	0.00	0.00	0.00	84.79
33453880	2013.10.18 01:05	buy	1.00	nzdjpy	83.009	0.000	0.000	2013.10.18 08:24	83.087	0.00	0.00	0.00	58.24
33454388	2013.10.18 01:15	buy	1.00	gbpnzd	1.90937	0.00000	0.00000	2013.10.18 07:33	1.91106	0.00	0.00	0.00	104.65
33469829	2013.10.18 07:37	buy	1.00	usdchf	0.90310	0.00000	0.00000	2013.10.21 13:57	0.90366	0.00	0.00	0.00	45.37
33469891	2013.10.18 07:39	buy	1.00	usdcad	1.02889	0.00000	0.00000	2013.10.21 02:41	1.02930	0.00	0.00	0.00	29.12
33469987	2013.10.18 07:44	buy	1.00	gbpchf	1.46026	0.00000	0.00000	2013.10.18 09:08	1.46039	0.00	0.00	0.00	10.53
33470024	2013.10.18 07:46	buy	1.00	nzdusd	0.84764	0.00000	0.00000	2013.10.18 08:24	0.84869	0.00	0.00	0.00	76.75
33473195	2013.10.18 08:40	buy	1.00	usdjpy	97.911	0.000	0.000	2013.10.20 23:59	97.942	0.00	0.00	0.00	23.13
33475978	2013.10.18 09:52	buy	1.00	eurnzd	1.61091	0.00000	0.00000	2013.10.18 11:49	1.61331	0.00	0.00	0.00	148.76
33480711	2013.10.18 11:54	buy	1.00	chfjpy	108.382	0.000	0.000	2013.10.18 17:19	108.547	0.00	0.00	0.00	123.19
33484173	2013.10.18 14:00	buy	1.00	eurusd	1.36794	0.00000	0.00000	2013.10.18 14:52	1.36949	0.00	0.00	0.00	113.18
33485294	2013.10.18 14:42	buy	1.00	gbpusd	1.61904	0.00000	0.00000	2013.10.21 14:29	1.61634	0.00	0.00	0.00	-197.49
33565998	2013.10.18 21:22	balance	Rollover - 33473195										0.10
33565999	2013.10.18 21:22	balance	Rollover - 33485294										0.60
33566000	2013.10.18 21:22	balance	Rollover - 33327342										0.40
33566001	2013.10.18 21:22	balance	Rollover - 33469829										0.40
33566002	2013.10.18 21:22	balance	Rollover - 33352280										-2.20
33566003	2013.10.18 21:22	balance	Rollover - 33469891										-2.20
33578917	2013.10.21 00:14	buy	1.00	gbpchf	1.45982	0.00000	0.00000	2013.10.21 07:23	1.46124	0.00	0.00	0.00	114.95
33578930	2013.10.21 00:15	buy	1.00	euraud	1.41631	0.00000	0.00000	2013.10.21 00:44	1.41735	0.00	0.00	0.00	73.38
33579310	2013.10.21 00:28	buy	1.00	audusd	0.96480	0.00000	0.00000	2013.10.21 02:21	0.96621	0.00	0.00	0.00	103.10
33579905	2013.10.21 00:45	buy	1.00	eurusd	1.36749	0.00000	0.00000	2013.10.21 09:43	1.36797	0.00	0.00	0.00	35.09
33582057	2013.10.21 02:30	buy	1.00	audusd	0.96612	0.00000	0.00000	2013.10.21 09:18	0.96728	0.00	0.00	0.00	84.85
33587382	2013.10.21 07:21	buy	1.00	nzdusd	0.84695	0.00000	0.00000	2013.10.22 10:41	0.84517	0.00	0.00	0.00	-130.20
33587922	2013.10.21 07:31	buy	1.00	euraud	1.41592	0.00000	0.00000	2013.10.21 17:42	1.41695	0.00	0.00	0.00	72.69
33588068	2013.10.21 07:33	buy	1.00	nzdcad	0.87239	0.00000	0.00000	2013.10.22 12:30	0.87337	0.00	0.00	0.00	69.29
33588226	2013.10.21 07:35	buy	1.00	nzdchf	0.76551	0.00000	0.00000	2013.10.22 12:45	0.76339	0.00	0.00	0.00	-171.84
33604136	2013.10.21 14:11	buy	1.00	eurusd	1.36624	0.00000	0.00000	2013.10.21 14:29	1.36727	0.00	0.00	0.00	75.33
33604166	2013.10.21 14:12	buy	1.00	gbpusd	1.61488	0.00000	0.00000	2013.10.21 14:29	1.61634	0.00	0.00	0.00	106.79
33604282	2013.10.21 14:17	buy	1.00	eurusd	1.36718	0.00000	0.00000	2013.10.21 17:40	1.36853	0.00	0.00	0.00	98.65
33604306	2013.10.21 14:18	buy	1.00	gbpusd	1.61560	0.00000	0.00000	2013.10.21 14:30	1.61651	0.00	0.00	0.00	66.55
33604727	2013.10.21 14:32	buy	1.00	gbpusd	1.61667	0.00000	0.00000	2013.10.22 12:30	1.61776	0.00	0.00	0.00	79.36
33604741	2013.10.21 14:33	buy	1.00	audusd	0.96742	0.00000	0.00000	2013.10.22 10:40	0.96646	0.00	0.00	0.00	-70.22
33604769	2013.10.21 14:34	buy	1.00	gbpjpy	158.688	0.000	0.000	2013.10.22 10:04	158.669	0.00	0.00	0.00	-14.13
33660650	2013.10.21 21:24	balance	Rollover - 33604727										0.60
33660651	2013.10.21 21:24	balance	Rollover - 33604741										2.90
33660652	2013.10.21 21:24	balance	Rollover - 33587382										3.00
33660653	2013.10.21 21:24	balance	Rollover - 33604769										1.00

33660654	2013.10.21 21:24	balance	Rollover - 33588226											2.90
33660655	2013.10.21 21:24	balance	Rollover - 33588068											1.80
33673079	2013.10.22 00:58	buy	1.00	eurpln	4.17580	4.16900	4.17785	2013.10.22 08:15	4.17785	0.00	0.00	0.00	49.07
33674259	2013.10.22 01:22	buy	1.00	cadjpy	95.332	0.000	0.000	2013.10.22 02:13	95.380	0.00	0.00	0.00	35.72
33674355	2013.10.22 01:25	buy	1.00	audjpy	94.843	0.000	0.000	2013.10.22 05:37	94.902	0.00	0.00	0.00	43.90
33676345	2013.10.22 02:22	buy	1.00	cadjpy	95.403	0.000	0.000	2013.10.22 05:34	95.426	0.00	0.00	0.00	17.11
33680561	2013.10.22 05:47	buy	1.00	gbpcad	1.66150	0.00000	0.00000	2013.10.22 12:30	1.66499	0.00	0.00	0.00	246.91
33688583	2013.10.22 10:06	buy	1.00	eurcad	1.40775	0.00000	0.00000	2013.10.22 12:30	1.41316	0.00	0.00	0.00	382.83
33688606	2013.10.22 10:07	buy	1.00	eurusd	1.36768	0.00000	0.00000	2013.10.22 12:33	1.37429	0.00	0.00	0.00	480.98
33688672	2013.10.22 10:09	buy	1.00	euraud	1.41488	0.00000	0.00000	2013.10.22 12:32	1.41724	0.00	0.00	0.00	166.52
33688781	2013.10.22 10:15	buy	1.00	gbpnzd	1.90799	0.00000	0.00000	2013.10.22 12:31	1.90755	0.00	0.00	0.00	-27.17
33692040	2013.10.22 12:27	buy	1.00	usdcad	1.03051	0.00000	0.00000	2013.10.22 12:31	1.02892	0.00	0.00	0.00	-112.43
33693851	2013.10.22 12:34	buy	1.00	audjpy	94.960	0.000	0.000	2013.10.22 12:34	94.965	0.00	0.00	0.00	3.72
33693893	2013.10.22 12:34	buy	1.00	eurgbp	0.84853	0.00000	0.00000	2013.10.22 12:56	0.84884	0.00	0.00	0.00	36.52
33693904	2013.10.22 12:34	buy	1.00	gbpchf	1.45374	0.00000	0.00000	2013.10.22 12:35	1.45426	0.00	0.00	0.00	42.16
33693927	2013.10.22 12:34	buy	1.00	audnzd	1.14191	0.00000	0.00000	2013.10.23 00:31	1.14376	0.00	0.00	0.00	114.30
33693963	2013.10.22 12:34	buy	1.00	euraud	1.41683	0.00000	0.00000	2013.10.22 13:11	1.41494	0.00	0.00	0.00	-133.57
33694000	2013.10.22 12:35	buy	1.00	gbpcad	1.66558	0.00000	0.00000	2013.10.22 12:40	1.66647	0.00	0.00	0.00	62.93
33694036	2013.10.22 12:35	buy	1.00	eurcad	1.41286	0.00000	0.00000	2013.10.22 12:37	1.41320	0.00	0.00	0.00	24.06
33694617	2013.10.22 12:41	buy	1.00	eurnzd	1.61435	0.00000	0.00000	2013.10.22 12:44	1.61510	0.00	0.00	0.00	46.44
33694644	2013.10.22 12:42	buy	1.00	cadchf	0.87222	0.00000	0.00000	2013.10.22 12:44	0.87271	0.00	0.00	0.00	39.71
33694933	2013.10.22 12:46	buy	1.00	chfjpy	109.131	0.000	0.000	2013.10.22 12:55	109.154	0.00	0.00	0.00	17.06
33695191	2013.10.22 12:49	buy	1.00	eurnzd	1.61615	0.00000	0.00000	2013.10.22 13:12	1.61475	0.00	0.00	0.00	-86.70
33696499	2013.10.22 13:03	buy	1.00	audnzd	1.14116	0.00000	0.00000	2013.10.23 00:31	1.14376	0.00	0.00	0.00	160.64
33696602	2013.10.22 13:04	buy	1.00	usdcad	1.02835	0.00000	0.00000	2013.10.22 13:05	1.02863	0.00	0.00	0.00	19.83
33697150	2013.10.22 13:09	buy	1.00	usdjpy	98.331	0.000	0.000	2013.10.22 13:10	98.350	0.00	0.00	0.00	14.07
33697754	2013.10.22 13:16	buy	1.00	usdjpy	98.438	0.000	0.000	2013.10.22 13:24	98.358	0.00	0.00	0.00	-59.21
33704459	2013.10.22 15:14	buy	1.00	usdchf	0.89479	0.00000	0.00000	2013.10.23 00:38	0.89485	0.00	0.00	0.00	4.87
33704513	2013.10.22 15:16	buy	1.00	gbpchf	1.45102	0.00000	0.00000	2013.10.22 16:29	1.45292	0.00	0.00	0.00	154.07
33704839	2013.10.22 15:26	buy	1.00	gbpnzd	1.90564	0.00000	0.00000	2013.10.22 16:46	1.90622	0.00	0.00	0.00	35.83
33707895	2013.10.22 16:31	buy	1.00	gbpchf	1.45286	0.00000	0.00000	2013.10.23 02:04	1.45333	0.00	0.00	0.00	38.11
33707930	2013.10.22 16:33	buy	1.00	cadchf	0.87050	0.00000	0.00000	2013.10.23 02:11	0.86949	0.00	0.00	0.00	-81.90
33707941	2013.10.22 16:33	buy	1.00	audchf	0.86888	0.00000	0.00000	2013.10.23 00:30	0.87137	0.00	0.00	0.00	201.93
33752589	2013.10.22 21:20	balance	Rollover - 33704459											0.40
33752590	2013.10.22 21:20	balance	Rollover - 33707895											1.20
33752591	2013.10.22 21:20	balance	Rollover - 33696499											-0.50
33752592	2013.10.22 21:20	balance	Rollover - 33693927											-0.50
33752593	2013.10.22 21:20	balance	Rollover - 33707941											3.30
33752594	2013.10.22 21:20	balance	Rollover - 33707930											1.40
33757744	2013.10.23 00:32	buy	1.00	nzdusd	0.85164	0.00000	0.00000	2013.10.23 02:11	0.84843	0.00	0.00	0.00	-232.88
33757763	2013.10.23 00:34	buy	1.00	audnzd	1.14462	0.00000	0.00000	2013.10.23 00:37	1.14519	0.00	0.00	0.00	35.19
33757764	2013.10.23 00:34	buy	1.00	euraud	1.41486	0.00000	0.00000	2013.10.23 01:44	1.41532	0.00	0.00	0.00	32.50
34063410	2013.10.25 08:19	buy	1.00	cadjpy	92.973	0.000	93.355	2013.10.25 10:21	93.085	0.00	0.00	0.00	83.49
34063660	2013.10.25 08:21	buy	1.00	audusd	0.95877	0.00000	0.96025	2013.10.25 11:21	0.95924	0.00	0.00	0.00	34.06
34064195	2013.10.25 08:26	buy	1.00	eurjpy	134.134	0.000	134.250	2013.10.25 08:45	134.194	0.00	0.00	0.00	44.71
34064685	2013.10.25 08:29	buy	1.00	eurchf	1.23223	0.00000	1.23225	2013.10.25 08:57	1.23227	0.00	0.00	0.00	3.25
34064941	2013.10.25 08:30	buy	1.00	eurgbp	0.85155	0.00000	0.00000	2013.10.25 08:46	0.85197	0.00	0.00	0.00	49.30

34065247	2013.10.25 08:32	buy	1.00	gbpchf	1.44789	0.00000	1.44791	2013.10.25 09:30	1.44792	0.00	0.00	0.00	2.43
34065601	2013.10.25 08:35	buy	1.00	gbpchf	1.44747	0.00000	1.44791	2013.10.25 09:30	1.44793	0.00	0.00	0.00	37.30
34066369	2013.10.25 08:43	buy	1.00	gbpchf	1.44640	0.00000	1.44791	2013.10.25 09:30	1.44792	0.00	0.00	0.00	123.25
34066781	2013.10.25 08:48	buy	1.00	eurgbp	0.85210	0.00000	0.85250	2013.10.25 14:06	0.85250	0.00	0.00	0.00	46.92
34074615	2013.10.25 10:29	buy	1.00	nzdusd	0.82965	0.00000	0.83065	2013.10.25 11:22	0.82991	0.00	0.00	0.00	18.84
34075119	2013.10.25 10:39	buy	1.00	euraud	1.43897	0.00000	1.43990	2013.10.25 14:47	1.43986	0.00	0.00	0.00	61.81
34075196	2013.10.25 10:40	buy	1.00	chfjpy	108.748	0.000	108.850	2013.10.25 11:22	108.801	0.00	0.00	0.00	39.47
34075331	2013.10.25 10:43	buy	1.00	audcad	1.00188	1.00115	1.00235	2013.10.25 11:11	1.00115	0.00	0.00	0.00	-50.67
34077665	2013.10.25 11:22	buy	1.00	eurusd	1.37986	0.00000	1.38075	2013.10.25 14:16	1.38057	0.00	0.00	0.00	51.43
34077688	2013.10.25 11:23	buy	1.00	gbpusd	1.61942	1.61760	1.64100	2013.10.28 05:20	1.61759	0.00	0.00	0.00	-132.64
34077706	2013.10.25 11:23	buy	1.00	usdjpy	97.329	97.225	97.375	2013.10.25 12:00	97.375	0.00	0.00	0.00	34.28
34077829	2013.10.25 11:25	buy	1.00	eurjpy	134.281	134.140	134.360	2013.10.25 12:13	134.139	0.00	0.00	0.00	-105.86
34077928	2013.10.25 11:25	buy	1.00	audusd	0.95932	0.95860	0.95995	2013.10.25 12:00	0.95858	0.00	0.00	0.00	-53.72
34079985	2013.10.25 12:02	buy	1.00	usdjpy	97.397	97.340	97.500	2013.10.25 12:07	97.340	0.00	0.00	0.00	-42.48
34080611	2013.10.25 12:15	buy	1.00	chfjpy	108.654	0.000	108.770	2013.10.25 12:59	108.741	0.00	0.00	0.00	64.85
34083301	2013.10.25 13:02	buy	1.00	audcad	1.00221	0.00000	1.00275	2013.10.25 13:31	1.00277	0.00	0.00	0.00	38.86
34084499	2013.10.25 13:56	buy	1.00	gbpcad	1.69063	1.68930	1.69130	2013.10.25 14:10	1.68929	0.00	0.00	0.00	-92.98
34085129	2013.10.25 14:16	buy	1.00	eurusd	1.38065	0.00000	1.38130	2013.10.27 22:10	1.38086	0.00	0.00	0.00	15.21
34085145	2013.10.25 14:17	buy	1.00	eurnzd	1.66151	0.00000	0.00000	2013.10.25 14:21	1.66197	0.00	0.00	0.00	27.68
34085253	2013.10.25 14:21	buy	1.00	eurnzd	1.66208	0.00000	0.00000	2013.10.25 14:42	1.66277	0.00	0.00	0.00	41.50
34085875	2013.10.25 14:45	buy	1.00	eurnzd	1.66321	0.00000	0.00000	2013.10.25 14:46	1.66424	0.00	0.00	0.00	61.89
34085922	2013.10.25 14:47	buy	1.00	eurnzd	1.66444	0.00000	1.66576	2013.10.25 15:08	1.66433	0.00	0.00	0.00	-6.61
34085926	2013.10.25 14:47	buy	1.00	euraud	1.43993	0.00000	1.44094	2013.10.25 15:51	1.44067	0.00	0.00	0.00	51.36
34086489	2013.10.25 15:00	buy	1.00	eurnzd	1.66333	0.00000	1.66576	2013.10.25 15:08	1.66437	0.00	0.00	0.00	62.49
34086909	2013.10.25 15:09	buy	1.00	gbpnzd	1.94997	0.00000	0.00000	2013.10.25 15:28	1.95142	0.00	0.00	0.00	87.05
34087131	2013.10.25 15:15	buy	1.00	eurnzd	1.66452	0.00000	0.00000	2013.10.25 15:26	1.66494	0.00	0.00	0.00	25.23
34088063	2013.10.25 15:52	buy	1.00	euraud	1.44057	0.00000	0.00000	2013.10.25 16:14	1.44104	0.00	0.00	0.00	32.62
34130859	2013.10.27 22:06	buy	1.00	audnzd	1.15751	1.15665	1.15894	2013.10.28 02:09	1.15665	0.00	0.00	0.00	-51.72
34143726	2013.10.28 01:12	buy	1.00	audusd	0.96047	0.00000	0.00000	2013.10.28 01:32	0.95989	0.00	0.00	0.00	-41.99
34143795	2013.10.28 01:13	buy	1.00	audusd	0.96044	0.00000	0.00000	2013.10.28 01:32	0.95987	0.00	0.00	0.00	-41.27
34143876	2013.10.28 01:14	buy	1.00	usdcad	1.04448	0.00000	1.04490	2013.10.28 14:07	1.04490	0.00	0.00	0.00	29.15
34147012	2013.10.28 01:45	buy	1.00	audjpy	93.699	0.000	0.000	2013.10.28 01:51	93.728	0.00	0.00	0.00	21.53
34149673	2013.10.28 02:15	buy	1.00	usdsgd	1.23657	1.23561	1.24000	2013.10.28 06:54	1.23561	0.00	0.00	0.00	-56.30
34171660	2013.10.28 06:59	buy	1.00	euraud	1.43758	0.00000	1.43902	2013.10.28 12:59	1.43902	0.00	0.00	0.00	100.07
34171855	2013.10.28 07:00	buy	1.00	gbpcad	1.68922	0.00000	1.68985	2013.10.28 07:53	1.68985	0.00	0.00	0.00	43.70
34172487	2013.10.28 07:06	buy	1.00	eurnzd	1.66350	0.00000	1.66521	2013.10.28 13:38	1.66452	0.00	0.00	0.00	61.28
34172808	2013.10.28 07:09	buy	1.00	usdpln	3.03311	0.00000	3.03680	2013.10.28 13:21	3.03426	0.00	0.00	0.00	27.49
34172917	2013.10.28 07:10	buy	1.00	usdpln	3.03231	0.00000	3.03680	2013.10.28 13:22	3.03446	0.00	0.00	0.00	51.39
34173192	2013.10.28 07:13	buy	1.00	gbpaud	1.68507	0.00000	1.68592	2013.10.28 08:00	1.68593	0.00	0.00	0.00	59.82
34173537	2013.10.28 07:17	buy	1.00	euraud	1.43767	0.00000	1.43902	2013.10.28 12:59	1.43903	0.00	0.00	0.00	94.51
34197627	2013.10.28 13:23	buy	1.00	eurjpy	134.711	0.000	134.808	2013.10.28 18:27	134.808	0.00	0.00	0.00	71.95
34197642	2013.10.28 13:23	buy	1.00	eurchf	1.23442	0.00000	0.00000	2013.10.28 17:22	1.23459	0.00	0.00	0.00	13.77
34197667	2013.10.28 13:24	buy	1.00	eurgbp	0.85357	0.85300	0.00000	2013.10.28 14:07	0.85299	0.00	0.00	0.00	-68.00
34197769	2013.10.28 13:27	buy	1.00	gbpcad	1.68707	0.00000	0.00000	2013.10.28 14:04	1.68780	0.00	0.00	0.00	50.69
34198440	2013.10.28 13:42	buy	1.00	eurnzd	1.66503	0.00000	0.00000	2013.10.28 13:54	1.66507	0.00	0.00	0.00	2.40
34198944	2013.10.28 13:54	buy	1.00	eurnzd	1.66533	0.00000	0.00000	2013.10.28 14:15	1.66363	0.00	0.00	0.00	-102.19
										0.00	0.00	0.00	6

246.58

Closed P/L: 6 246.58

Open Trades:

Ticket	Open Time	Type	Size	Item	Price	S / L	T / P	Price	Commission	Taxes	Swap	Profit
34198670	2013.10.28 13:47	buy	1.00	audjpy	93.543	0.000	0.000	93.516	0.00	0.00	0.00	-20.03
34197746	2013.10.28 13:26	buy	1.00	audnzd	1.15581	0.00000	0.00000	1.15363	0.00	0.00	0.00	-131.11
34197574	2013.10.28 13:20	buy	1.00	audusd	0.95822	0.00000	0.00000	0.95792	0.00	0.00	0.00	-21.73
34216166	2013.10.28 18:59	buy	1.00	eurjpy	134.867	0.000	0.000	134.772	0.00	0.00	0.00	-70.49
34184827	2013.10.28 09:48	buy	1.00	eurusd	1.38123	0.00000	0.00000	1.38051	0.00	0.00	0.00	-52.15
34197511	2013.10.28 13:19	buy	1.00	gbpusd	1.61595	0.00000	0.00000	1.61575	0.00	0.00	0.00	-14.49
									0.00	0.00	0.00	-310.00

Floating P/L: -310.00

Attachment II : Indicator Codes

FXCM Multiple Currency Overview Indicator Code

```
//+------------------------------------------------------------------+
//|                        FXCM Multiple Currency Overview.mq4 |
//|                                                  Craig Nesbitt |
//|                                                canesbitt@wpi.edu |
//+------------------------------------------------------------------+
#property copyright "Craig Nesbitt"
#property link      "canesbitt@wpi.edu"
#property indicator_chart_window

// Currency Period Matrix Definitions
#define CHART_TitleCount 8
#define CHART_currencyPairCount 8
#define CHART_periodCount 9
#define CHART_periods 8

// Strength Definitions
#define STR_CURRENCIES 8
#define STR_PAIRS 28
#define Strengths_Displayed 28

// Global MA Analysis Definitions
#define pairs 28

/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                                CHART VARIABLES                                    START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/

string currencyTitles[CHART_TitleCount] = {"USD","EUR","JPY","AUD","CAD","GBP","CHF","NZD"};

/*
string
CHARTpairNames[]={"USDEUR","USDJPY","USDAUD","USDCAD","USDGBP","USDCHF","USDNZD","
STR",  //USD
                  "EURUSD","EURJPY","EURAUD","EURCAD","EURGBP","EURCHF","EURNZD","   STR",
//EUR
                  "JPYUSD","JPYEUR","JPYAUD","JPYCAD","JPYGBP","JPYCHF","JPYNZD","   STR",  //JPY
                  "AUDUSD","AUDEUR","AUDJPY","AUDCAD","AUDGBP","AUDCHF","AUDNZD","   STR",
//AUD
                  "CADUSD","CADEUR","CADJPY","CADAUD","CADGBP","CADCHF","CADNZD","   STR",
//CAD
                  "GBPUSD","GBPEUR","GBPJPY","GBPCAD","GBPAUD","GBPCHF","GBPNZD","   STR",
//GBP
                  "CHFUSD","CHFEUR","CHFJPY","CHFCAD","CHFAUD","CHFGBP","CHFNZD","   STR",
//CHF
```

```
            "NZDUSD","NZDEUR","NZDJPY","NZDCAD","NZDAUD","NZDGBP","NZDCHF","   STR"};
//NZD
*/

string CHART_pairNames[]={"   EUR","   JPY","   AUD","   CAD","   GBP","   CHF","   NZD","  TOTAL",
//USD
                  "   USD","   JPY","   AUD","   CAD","   GBP","   CHF","   NZD","  TOTAL",  //EUR
                  "   USD","   EUR","   AUD","   CAD","   GBP","   CHF","   NZD","  TOTAL",  //JPY
                  "   USD","   EUR","   JPY","   CAD","   GBP","   CHF","   NZD","  TOTAL",  //AUD
                  "   USD","   EUR","   JPY","   AUD","   GBP","   CHF","   NZD","  TOTAL",  //CAD
                  "   USD","   EUR","   JPY","   CAD","   AUD","   CHF","   NZD","  TOTAL",  //GBP
                  "   USD","   EUR","   JPY","   CAD","   AUD","   GBP","   NZD","  TOTAL",  //CHF
                  "   USD","   EUR","   JPY","   CAD","   AUD","   GBP","   CHF","  TOTAL"}; //NZD


string USDpairs [CHART_currencyPairCount]    = {"EURUSD", "USDJPY", "AUDUSD", "USDCAD",
"GBPUSD", "USDCHF", "NZDUSD" };
string EURpairs [CHART_currencyPairCount]    = {"EURUSD", "EURJPY", "EURAUD", "EURCAD",
"EURGBP", "EURCHF", "EURNZD" };
string JPYpairs [CHART_currencyPairCount]    = {"USDJPY", "EURJPY", "AUDJPY", "CADJPY",
"GBPJPY", "CHFJPY", "NZDJPY" };
string AUDpairs [CHART_currencyPairCount]    = {"AUDUSD", "EURAUD", "AUDJPY", "AUDCAD",
"GBPAUD", "AUDCHF", "AUDNZD" };
string CADpairs [CHART_currencyPairCount]    = {"USDCAD", "EURCAD", "CADJPY", "AUDCAD",
"GBPCAD", "CADCHF", "NZDCAD" };
string GBPpairs [CHART_currencyPairCount]    = {"GBPUSD", "EURGBP", "GBPJPY", "GBPCAD",
"GBPAUD", "GBPCHF", "GBPNZD" };
string CHFpairs [CHART_currencyPairCount]    = {"USDCHF", "EURCHF", "CHFJPY", "CADCHF",
"AUDCHF", "GBPCHF", "NZDCHF" };
string NZDpairs [CHART_currencyPairCount]    = {"NZDUSD", "EURNZD", "NZDJPY", "NZDCAD",
"AUDNZD", "GBPNZD", "NZDCHF" };


string periodTitles[CHART_periodCount] = {  "M5"  ,   "M15"  ,  "M30"   ,  "H1"   ,  "H4"   ,  "D1"   ,
"W1"   ,    "MN" ,   "TRD"  };
double   periodValues[CHART_periods] = {PERIOD_M5, PERIOD_M15, PERIOD_M30, PERIOD_H1,
PERIOD_H4, PERIOD_D1, PERIOD_W1, PERIOD_MN1};
double   USDStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   EURStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   JPYStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   AUDStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   CADStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   GBPStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   CHFStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};
double   NZDStrengths[CHART_periods] = {   0,        0,        0,        0,        0,        0,        0,        0
};


int   NewLineValue   = 3; // Sets number of charts shown accross a row before making a new line
```

```
//chart display controls
extern int column_width  =56,  // horizontal interval at which the squares are created
         row_height    =19,  // vertical interval
         column_offset =40,  // horizontal indent of all squares
         row_offset    =60,  // vertical indent
         fontSize      =10,  // size of text
         numberSize    =11;  // size of numbers

//chart timeframe controls
extern int barsBack = 0;   // displays number of bars back in the history the indicator reports

/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
END                                CHART VARIABLES                                        END
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/




/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                              STRENGTH VARIABLES                                     START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/



// WEIGHT RATIO
extern int rank_weight = 1;
extern int positives_weight = 1;


// Content Properties
extern int History_Shift = 0;   // number of bars in the history that the indicator is calculating

extern color TitleColor = White;         // Color corresponding to currency name
extern color Number1    = Red;           // Color corresponding to 1
extern color Number2    = Maroon;        // Color corresponding to 2
extern color Number3    = Chocolate;     // Color corresponding to 3
extern color Number4    = DarkGoldenrod; // Color corresponding to 4
extern color Number5    = Olive;         // Color corresponding to 5
extern color Number6    = OliveDrab;     // Color corresponding to 6
extern color Number7    = Green;         // Color corresponding to 7
extern color Number8    = Lime;          // Color corresponding to 8


/////////////// GLOBAL VARIABLES ///////////////

string   titleFont = "Arial Bold";   // Default Title font
string   monospace = "Courier New";  // Default font style
```

```
int tickvolume = 0; // used to determine a new bar.

// list of currencies being measured
// this is the array that will be printed as titles
string currencynames[STR_CURRENCIES] = {"USD",  // pos = 0
                                        "EUR",  // pos = 1
                                        "JPY",  // pos = 2
                                        "AUD",  // pos = 3
                                        "CAD",  // pos = 4
                                        "GBP",  // pos = 5
                                        "CHF",  // pos = 6
                                        "NZD"}; // pos = 7



// list of all pairnames for currencies
// used to calculate pipsum for each currency
string pairnames[STR_PAIRS] = {"EURUSD", "GBPUSD", "USDCHF", "USDJPY", "AUDUSD",
"EURCHF",
                               "EURGBP", "NZDUSD", "USDCAD", "AUDCAD", "AUDCHF", "CADJPY",
                               "CHFJPY", "EURAUD", "EURCAD", "EURNZD", "GBPAUD", "GBPCAD",
                               "GBPCHF", "GBPNZD", "CADCHF", "NZDCHF", "NZDCAD", "NZDJPY",
                               "AUDNZD", "AUDJPY", "EURJPY", "GBPJPY"};

// strength arrays for each currency for each bar. maximum 100 bars
// this is the array that will be printed for each currency
int USDstrengths [Strengths_Displayed];
int EURstrengths [Strengths_Displayed];
int JPYstrengths [Strengths_Displayed];
int AUDstrengths [Strengths_Displayed];
int CADstrengths [Strengths_Displayed];
int GBPstrengths [Strengths_Displayed];
int CHFstrengths [Strengths_Displayed];
int NZDstrengths [Strengths_Displayed];


// sum of pips for each currency for each bar. maximum 100 bars
// used to rank the currencies (1-8) by pip change
double USDpipsum [Strengths_Displayed];
double EURpipsum [Strengths_Displayed];
double JPYpipsum [Strengths_Displayed];
double AUDpipsum [Strengths_Displayed];
double CADpipsum [Strengths_Displayed];
double GBPpipsum [Strengths_Displayed];
double CHFpipsum [Strengths_Displayed];
double NZDpipsum [Strengths_Displayed];

// rank of each currency (1-8) for each bar. maximum 100 bars
// used in part for determining final strength
int USDranked [Strengths_Displayed];
int EURranked [Strengths_Displayed];
int JPYranked [Strengths_Displayed];
int AUDranked [Strengths_Displayed];
int CADranked [Strengths_Displayed];
int GBPranked [Strengths_Displayed];
```

```
int CHFranked [Strengths_Displayed];
int NZDranked [Strengths_Displayed];

// count of positive pips for each each bar. maximum 100 bars
// used in part for determining final strength
int USDpositivescount [Strengths_Displayed];
int EURpositivescount [Strengths_Displayed];
int JPYpositivescount [Strengths_Displayed];
int AUDpositivescount [Strengths_Displayed];
int CADpositivescount [Strengths_Displayed];
int GBPpositivescount [Strengths_Displayed];
int CHFpositivescount [Strengths_Displayed];
int NZDpositivescount [Strengths_Displayed];

// arrays to hold ranked currency name and pipsum for the most recent bar
// used to assign rank number to currenyranked
double rankedpipsum   [STR_CURRENCIES];
string rankednames    [STR_CURRENCIES] = {"USD", "EUR", "JPY", "AUD",
                                          "CAD", "GBP", "CHF", "NZD"};

// holds currency information for the current bar.
int currencyrank [STR_CURRENCIES];
double currencypipsum [STR_CURRENCIES];



/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
END                                STRENGTH VARIABLES                                END
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/



/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                        CURRENCY PAIR INFORMATION ARRAYS                        START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/

/*
// Currency pair ID's of the 8 major currencies
string pairnames[pairs] = {"EURUSD", // pos = 0
                           "GBPUSD", // pos = 1
                           "USDCHF", // pos = 2
                           "USDJPY", // pos = 3
                           "AUDUSD", // pos = 4
                           "EURCHF", // pos = 5
                           "EURGBP", // pos = 6
                           "NZDUSD", // pos = 7
                           "USDCAD", // pos = 8
                           "AUDCAD", // pos = 9
                           "AUDCHF", // pos = 10
                           "CADJPY", // pos = 11
                           "CHFJPY", // pos = 12
                           "EURAUD", // pos = 13
```

```
                        "EURCAD", // pos = 14
                        "EURNZD", // pos = 15
                        "GBPAUD", // pos = 16
                        "GBPCAD", // pos = 17
                        "GBPCHF", // pos = 18
                        "GBPNZD", // pos = 19
                        "CADCHF", // pos = 20
                        "NZDCHF", // pos = 21
                        "NZDCAD", // pos = 22
                        "NZDJPY", // pos = 23
                        "AUDNZD", // pos = 24
                        "AUDJPY", // pos = 25
                        "EURJPY", // pos = 26
                        "GBPJPY"};// pos = 27
*/

// array of the number of alerts  in the currenct bar corresponding to pairnames[] positions
int alertCount[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                          0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                          0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

// array of time of last alert corresponding to pairnames[] positions
int alertTime[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                         0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                         0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

// array of opening time of the bar of the last alert corresponding to pairnames[] positions
int alertBar[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                        0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                        0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

// array of alert type of last alert corresponding to pairnames[] positions
//    -3 corresponds to an initialization sell order.
//    -2 corresponds to a correctional sell order.
//    -1 corresponds to sell
//     1 corresponds to buy
//     2 corresponds to a correctional buy.
//     3 corresponds to an initialization buy order.
//     0 corresponds to Close Position
int alertType[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                         0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                         0, 0, 0, 0, 0, 0, 0, 0, 0, 0};

// array of alert type of most recent Opening order corresponding to pairnames[] positions
//   -3 corresponds to an initialization sell order.
//   -2 corresponds to a correctional sell order.
//   -1 corresponds to a sell order.
//    1 corresponds to a buy order.
//    2 corresponds to a correctional buy order.
//    3 corresponds to an initialization buy order.
//    0 means there has not been an order
int openAlert[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                         0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                         0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
```

```
// array of tracking status corresponding to pairnames[] positions
//    1 Tracks the currency pair
//    0 Ignores the currency pair
// array is constructed by buildTrackCurrency() helper function
int trackCurrency[pairs];


// array of the volume of the current bar corresponding to pairnames[] positions
//    used to determine if the indicator has started a new bar.
int currencyVolumes[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                               0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                               0, 0, 0, 0, 0, 0, 0, 0};



// array of the Spread of the currency pair corresponding to pairnames[] positions
double currencySpreads[pairs];

//array of the last error type for the currency corresponding to pairnames[] positions
int    history_error[pairs] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
                                0, 0, 0, 0, 0, 0, 0, 0};

/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
END                          CURRENCY PAIR INFORMATION ARRAYS                          END
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/



/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                          GMAA EXTERNAL VARIABLES                          START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/


/*extern*/ bool  DISPLAY_ALERTS    = true;       // Determines whether or not alerts will be displayed.


            // MOVING AVERAGE VARIABLES
extern int    SMA            = 50;          // Simple moving average tracer
extern int    EMA1           = 5;           // first exp. moving average
extern int    EMA2           = 20;          // second exp. moving average

            // GRAPH AND POSITION VARIABLES
extern int    HISTORY_SHIFT   = 0;           // horizontal shift of the indicator action

            // SAFE-GUARD VARIABLES
extern int    ALERTS_PER_BAR     = 0;        // Limit of alerts prompted per bar
extern int    WAIT_AFTER_ALERT   = 300;      // Number of seconds to wait until reanalyzing bar
extern double BUY_CROSS_BUFFER   = 3;        // Number of pips beyond cross to confirm  a buy alert
extern double SELL_CROSS_BUFFER  = 3;        // Number of pips beyond cross to confirm  a sell alert
```

```
extern double CLOSE_CROSS_BUFFER = 3;         // Number of pips beyond cross to confirm  a close-
posiition alert
extern double GAP_MARGIN_OF_ERROR = 0.1;      // Margin of error allowed when checking for gaps in
history

          // INITIALIZATION VARIABLES
extern int    INITIALIZE_BARS    = 3;       // Number of pips between MAs to confirm initialize alerts


          // CURRENCY TRACKING OPTIONS
          //    - TRACK  THE CURRENCY (1)
          //    - IGNORE THE CURRENCY (0)
extern bool   Track_EURUSD    = 1,           // (track/ignore) EURUSD currency pair
          Track_GBPUSD    = 1,           // (track/ignore) GBPUSD currency pair
          Track_USDCHF    = 1,           // (track/ignore) USDCHF currency pair
          Track_USDJPY    = 1,           // (track/ignore) USDJPY currency pair
          Track_AUDUSD    = 1,           // (track/ignore) AUDUSD currency pair
          Track_EURCHF    = 1,           // (track/ignore) EURCHF currency pair
          Track_EURGBP    = 1,           // (track/ignore) EURGBP currency pair
          Track_NZDUSD    = 1,           // (track/ignore) NZDUSD currency pair
          Track_USDCAD    = 1,           // (track/ignore) USDCAD currency pair
          Track_AUDCAD    = 1,           // (track/ignore) AUDCAD currency pair
          Track_AUDCHF    = 1,           // (track/ignore) AUDCHF currency pair
          Track_CADJPY    = 1,           // (track/ignore) CADJPY currency pair
          Track_CHFJPY    = 1,           // (track/ignore) CHFJPY currency pair
          Track_EURAUD    = 1,           // (track/ignore) EURAUD currency pair
          Track_EURCAD    = 1,           // (track/ignore) EURCAD currency pair
          Track_EURNZD    = 1,           // (track/ignore) EURNZD currency pair
          Track_GBPAUD    = 1,           // (track/ignore) GBPAUD currency pair
          Track_GBPCAD    = 1,           // (track/ignore) GBPCAD currency pair
          Track_GBPCHF    = 1,           // (track/ignore) GBPCHF currency pair
          Track_GBPNZD    = 1,           // (track/ignore) GBPNZD currency pair
          Track_CADCHF    = 1,           // (track/ignore) CADCHF currency pair
          Track_NZDCHF    = 1,           // (track/ignore) NZDCHF currency pair
          Track_NZDCAD    = 1,           // (track/ignore) NZDCAD currency pair
          Track_NZDJPY    = 1,           // (track/ignore) NZDJPY currency pair
          Track_AUDNZD    = 1,           // (track/ignore) AUDNZD currency pair
          Track_AUDJPY    = 1,           // (track/ignore) AUDJPY currency pair
          Track_EURJPY    = 1,           // (track/ignore) EURJPY currency pair
          Track_GBPJPY    = 1,           // (track/ignore) GBPJPY currency pair

          Trade_on_Sunday    = 1,        // (allow/restrict) opening trades on Sunday
          Trade_on_Monday    = 1,        // (allow/restrict) opening trades on Monday
          Trade_on_Tuesday   = 1,        // (allow/restrict) opening trades on Tuesday
          Trade_on_Wednesday = 1,        // (allow/restrict) opening trades on Wednesday
          Trade_on_Thursday  = 1,        // (allow/restrict) opening trades on Thursday
          Trade_on_Friday    = 1,        // (allow/restrict) opening trades on Friday
          Trade_on_Saturday  = 1;        // (allow/restrict) opening trades on Saturday




          // TIME WINDOW TO ALLOW TRADING
          // WHERE INT IS HOUR-OF-DAY
extern datetime   Sunday_Open     = D'00:00:00',  // Open of Sunday trading window
```

```
Sunday_Close    = D'23:59:59',  // Close of Sunday trading window

Monday_Open     = D'00:00:00',  // Open of Monday trading window
Monday_Close    = D'23:59:59',  // Close of Monday trading window

Tuesday_Open    = D'00:00:00',  // Open of Tuesday trading window
Tuesday_Close   = D'23:59:59',  // Close of Tuesday trading window

Wednesday_Open  = D'00:00:00',  // Open of Wednesday trading window
Wednesday_Close = D'23:59:59',  // Close of Wednesday trading window

Thursday_Open   = D'00:00:00',  // Open of Thursday trading window
Thursday_Close  = D'23:59:59',  // Close of Thursday trading window

Friday_Open     = D'00:00:00',  // Open of Friday trading window
Friday_Close    = D'23:59:59',  // Close of Friday trading window

Saturday_Open   = D'00:00:00',  // Open of Saturday trading window
Saturday_Close  = D'23:59:59';  // Close of Saturday trading window


/*********************************************************************************************
---------------------------------------------------------------------------------------------
END                         GMAA EXTERNAL VARIABLES                          END
---------------------------------------------------------------------------------------------
*********************************************************************************************/




/*********************************************************************************************
---------------------------------------------------------------------------------------------
START                         GMAA GLOBAL VARIABLES                          START
---------------------------------------------------------------------------------------------
*********************************************************************************************/

// Indicates what condition was met to generate the alert.
string AlertCausedBy;

int inputError = 0;

/*********************************************************************************************
---------------------------------------------------------------------------------------------
END                         GMAA GLOBAL VARIABLES                          END
---------------------------------------------------------------------------------------------
*********************************************************************************************/




/*********************************************************************************************
```

```
----------------------------------------------------------------------------------------------------
START                                MAIN FUNCTIONS                                START
----------------------------------------------------------------------------------------------------
****************************************************************************************************/

//+------------------------------------------------------------------+
//| Custom indicator initialization function                         |
//+------------------------------------------------------------------+
int init()
  {
   string short_name = "FXCM Multiple Currency Overview";
   IndicatorShortName(short_name);
   CPM_buildMatrix(); //creates currency pair matrix
   buildTrackCurrency();
   return(0);
  }



//+------------------------------------------------------------------+
//| Custom indicator deinitialization function                       |
//+------------------------------------------------------------------+
int deinit()
{
   ObjectsDeleteAll();
   return(0);
}

//+------------------------------------------------------------------+
//| Custom indicator iteration function                              |
//+------------------------------------------------------------------+
int start()
  {
   int    counted_bars=IndicatorCounted();
//----
   alertTheWorld(); // checks all currencies for an alert.
//----
   return(0);
  }
//+------------------------------------------------------------------+


/****************************************************************************************************
----------------------------------------------------------------------------------------------------
END                                  MAIN FUNCTIONS                                END
----------------------------------------------------------------------------------------------------
****************************************************************************************************/




/****************************************************************************************************
----------------------------------------------------------------------------------------------------
START                                CHART FUNCTIONS                               START
----------------------------------------------------------------------------------------------------
```

```
**********************************************************************************************************/

//+------------------------------------------------------------------+
//| Custom indicator CPM_buildMatrix function                        |
//+------------------------------------------------------------------+
void CPM_buildMatrix()
{
  int x,y;

  int BeginChartX,
     BeginChartY,
     BeginChartX_count = 0,
     BeginChartY_count = 0,
     TitlePlaceY,
     TitlePlaceX,
     CurrentTitle = 0,
     CurrentX = 0,
     currentChart = 0;

  int ChartRows;
  if (CHART_TitleCount%NewLineValue == 0)
     ChartRows = CHART_TitleCount/NewLineValue;
  else
     ChartRows = CHART_TitleCount/NewLineValue + 1;


  //creates titles for each set
  for (y=0; y < ChartRows; y++)
  {
    BeginChartX_count = 0;

    BeginChartY = BeginChartY_count*(CHART_periodCount*row_height + 1.5*row_offset);
    TitlePlaceY = row_offset - 4/5.0*row_offset + BeginChartY;

    for (x=0; x < NewLineValue; x++)
    {
      if (StringLen(currencyTitles[CurrentTitle]) != 0)
      {
        BeginChartX = BeginChartX_count *(CHART_currencyPairCount*column_width +
2*column_offset);
        TitlePlaceX = column_offset*1.5 + column_offset*CHART_currencyPairCount/2.0 + BeginChartX;

        ObjectCreate("CurrencyTitle"+x+y,OBJ_LABEL,0,0,0,0,0);
        ObjectSet("CurrencyTitle"+x+y,OBJPROP_XDISTANCE,TitlePlaceX);
        ObjectSet("CurrencyTitle"+x+y,OBJPROP_YDISTANCE,TitlePlaceY);
        ObjectSetText("CurrencyTitle"+x+y, currencyTitles[CurrentTitle],fontSize*1.5,"Courier
New",Khaki);


        // Y axis Label
        for (int i=0;i < CHART_periodCount;i++)
        {
          ObjectCreate("Period"+x+y+i,OBJ_LABEL,0,0,0,0,0);
          ObjectSet("Period"+x+y+i,OBJPROP_XDISTANCE,column_offset/3.0 + BeginChartX);
          ObjectSet("Period"+x+y+i,OBJPROP_YDISTANCE,i*row_height + row_offset + BeginChartY);
          ObjectSetText("Period"+x+y+i,periodTitles[i],fontSize,"Courier New",White);
```

```
         }

         // X axis Label
         for (int k=0;k < CHART_currencyPairCount;k++)
         {
           ObjectCreate("CurrencyPair"+x+y+k,OBJ_LABEL,0,0,0,0,0); // creates the label
           ObjectSet("CurrencyPair"+x+y+k,OBJPROP_XDISTANCE,k*column_width+column_offset +
BeginChartX); // label x position
           ObjectSet("CurrencyPair"+x+y+k,OBJPROP_YDISTANCE,row_offset -20 + BeginChartY); //
label y position
           ObjectSetText("CurrencyPair"+x+y+k,CHART_pairNames[CurrentX],fontSize,"Courier
New",White); //label text, font, and color
           CurrentX++;
         }

       }

       BeginChartX_count++;
       CurrentTitle++;

     }

     BeginChartY_count++;
  }
  CurrentTitle = 0;
  BeginChartX_count = 0;
  BeginChartY_count = 0;
  BeginChartX = 0;
  BeginChartY = 0;


 // draws the boxes as matrix contents
  for(x=0; x < NewLineValue * CHART_currencyPairCount; x++)
  {
    //resets Y position
    BeginChartY_count = 0;

    //used to offset X for set.
    if (x>0 && x%CHART_currencyPairCount == 0)
      BeginChartX_count++;

    //offsets X for set
    BeginChartX = BeginChartX_count* 2*column_offset;

    for(y=0;y < ChartRows*CHART_periodCount ;y++)
    {
      //calculates the current subchart being filled
      currentChart = 3*(y/CHART_periodCount) + x/CHART_currencyPairCount +1;

      // If subchart being filled is less than total subcharts
      if (currentChart <= CHART_TitleCount)
      {
        //used to offset Y for set.
        if (y>0 && y % CHART_periodCount == 0)
          BeginChartY_count++;
```

```
         //offsets Y for set.
         BeginChartY = BeginChartY_count*(1.5*row_offset);

         // Contents
         ObjectCreate("PairValue"+(x*100)+y,OBJ_LABEL,0,0,0,0,0);
         ObjectSet("PairValue"+(x*100)+y,OBJPROP_XDISTANCE,x*column_width +column_offset +
BeginChartX);
         ObjectSet("PairValue"+(x*100)+y,OBJPROP_YDISTANCE,y*row_height+row_offset +
BeginChartY);
         ObjectSetText("PairValue"+(x*100)+y,CharToStr(110),fontSize,"Wingdings",White);
       }
     }

     if (x == CHART_currencyPairCount)
       CurrentTitle+=NewLineValue;
   }

   //offsets X for set
   BeginChartX = BeginChartX_count * 9.25 *column_width;
   //offsets Y for set.
   BeginChartY = BeginChartY_count *(27*row_height);



   // STRENGTH HISTORY IN BOTTOM RIGHT CORNER
   ObjectCreate("Strength History", OBJ_LABEL,0,0,0,0,0);
   ObjectSet   ("Strength History", OBJPROP_XDISTANCE, BeginChartX + 3*column_width +
column_offset);
   ObjectSet   ("Strength History", OBJPROP_YDISTANCE, BeginChartY - 1.5* row_height + row_offset);
   ObjectSetText("Strength History", "Strength History",fontSize,"Courier New",Khaki);

   for (int title = 0; title < STR_CURRENCIES; title++) {
     ObjectCreate (currencynames[title], OBJ_LABEL,0,0,0,0,0);                 // Creates symbol label
     ObjectSet    (currencynames[title], OBJPROP_XDISTANCE, column_offset - 1/3.0*column_width +
BeginChartX);                          // Places symbol X coordinate
     ObjectSet    (currencynames[title], OBJPROP_YDISTANCE,title*row_height + row_offset +
BeginChartY);           // Places symbol Y coordinate
     ObjectSetText(currencynames[title], currencynames[title],fontSize-2,titleFont,TitleColor);  // Sets
symbol Font

     for (int str = 1; str <= Strengths_Displayed; str++) {
       //Currency pair information (fills graph)
       ObjectCreate (currencynames[title] + str, OBJ_LABEL,0,0,0,0,0);             // Creates graph contents
label
       ObjectSet    (currencynames[title] + str, OBJPROP_XDISTANCE, str*2/7.0*column_width +
column_offset + BeginChartX);         // Places graph contents X
       ObjectSet    (currencynames[title] + str, OBJPROP_YDISTANCE,title*row_height + row_offset +
BeginChartY);        // Places graph contents Y
       ObjectSetText(currencynames[title] + str, CharToStr(110),fontSize,"WingDings",TitleColor);   // Sets
graph contents Font
     }
   }
   fillStrength();
}
```

```
//+------------------------------------------------------------------+
//| Custom indicator CPM_updateMatrix function                       |
//+------------------------------------------------------------------+
void CPM_updateMatrix()
{
  int  x_position = 0; // RESETS POSITION FOR UPDATE
  int  y_position = 0; // RESETS POSITION FOR UPDATE
  int  ChartCount = 0; // DECLARES CHART POSITION
  double PipChange;
  bool isInvertedPair;
////////////////////////////////////////////////////////////////  USD

  for (int k=0; k<CHART_periodCount-1; k++)
    USDStrengths[k] = 0;

  for (int i=0; i< CHART_currencyPairCount; i++)
  {
    if (StringFind(USDpairs[i],"USD",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;               // then run inverted function

    else
      isInvertedPair = false;


    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
            PipChange = CPM_update_cell_inverted (USDpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(USDpairs[i], periodValues[k], barsBack, x_position,
y_position);

          USDStrengths[k] = CPM_addToStrength(USDStrengths[k], PipChange);
        }

        else CPM_alertCondition(USDpairs[i], isInvertedPair, x_position, y_position);

        y_position++;

      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(USDStrengths[k], x_position, y_position);
          y_position++;
```

```
        }

        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }

    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }

////////////////////////////////////////////////////////////////  EUR

  // NEW CHART
  ChartCount++;
    if (ChartCount%NewLineValue == 0)
      x_position = 0;
    else
      x_position++;

  y_position = ChartCount/NewLineValue* CHART_periodCount;

  for (k=0; k<CHART_periodCount-1; k++)
    EURStrengths[k] = 0;

  for (i=0; i<CHART_currencyPairCount; i++)
  {
    if (StringFind(EURpairs[i],"EUR",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;               // then run inverted function

    else
      isInvertedPair = false;

    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
            PipChange = CPM_update_cell_inverted (EURpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(EURpairs[i], periodValues[k], barsBack, x_position,
y_position);

          EURStrengths[k] = CPM_addToStrength(EURStrengths[k], PipChange);
        }

        else CPM_alertCondition(EURpairs[i], isInvertedPair, x_position, y_position);

        y_position++;
```

```
    }
  }

  if ( i == CHART_currencyPairCount-1)
  {
    for (k=0; k<CHART_periodCount; k++)
    {
      if( k < CHART_periodCount-1)
      {
        CPM_printStrength(EURStrengths[k], x_position, y_position);
        y_position++;
      }

      else ObjectDelete("PairValue"+ (x_position*100) + y_position);
    }
  }

  if (i != CHART_currencyPairCount-1)
  {
    x_position++;
    y_position = ChartCount/NewLineValue* CHART_periodCount;
  }
}



////////////////////////////////////////////////////////////////  jpy

// NEW CHART
ChartCount++;
  if (ChartCount%NewLineValue == 0)
    x_position = 0;
  else
    x_position++;

y_position = ChartCount/NewLineValue* CHART_periodCount;

for (k=0; k<CHART_periodCount-1; k++)
  JPYStrengths[k] = 0;

for (i=0; i<CHART_currencyPairCount; i++)
{
  if (StringFind(JPYpairs[i],"JPY",0) > 0) // if the currency is not at beginning of pair
    isInvertedPair = true;               // then run inverted function

  else
    isInvertedPair = false;

  if ( i< CHART_currencyPairCount-1)
  {
    for (k=0; k<CHART_periodCount; k++)
    {
      if( k < CHART_periodCount-1)
      {
        if (isInvertedPair)
```

```
            PipChange = CPM_update_cell_inverted (JPYpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(JPYpairs[i], periodValues[k], barsBack, x_position,
y_position);

          JPYStrengths[k] = CPM_addToStrength(JPYStrengths[k], PipChange);
        }

        else CPM_alertCondition(JPYpairs[i], isInvertedPair, x_position, y_position);

        y_position++;

      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(JPYStrengths[k], x_position, y_position);
          y_position++;
        }

        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }

    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }




////////////////////////////////////////////////////////////////////  AUD

  // NEW CHART
  ChartCount++;
    if (ChartCount%NewLineValue == 0)
      x_position = 0;
    else
      x_position++;

  y_position = ChartCount/NewLineValue* CHART_periodCount;


  for (k=0; k<CHART_periodCount-1; k++)
    AUDStrengths[k] = 0;
```

```
  for (i=0; i<CHART_currencyPairCount; i++)
  {
    if (StringFind(AUDpairs[i],"AUD",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;                 // then run inverted function

    else
      isInvertedPair = false;

    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
            PipChange = CPM_update_cell_inverted (AUDpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(AUDpairs[i], periodValues[k], barsBack, x_position,
y_position);

          AUDStrengths[k] = CPM_addToStrength(AUDStrengths[k], PipChange);
        }

        else CPM_alertCondition(AUDpairs[i], isInvertedPair, x_position, y_position);

        y_position++;

      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(AUDStrengths[k], x_position, y_position);
          y_position++;
        }

        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }


    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }
```

```
////////////////////////////////////////////////////////////////  CAD

  // NEW CHART
  ChartCount++;
    if (ChartCount%NewLineValue == 0)
      x_position = 0;
    else
      x_position++;

  y_position = ChartCount/NewLineValue* CHART_periodCount;

  for (k=0; k<CHART_periodCount-1; k++)
    CADStrengths[k] = 0;

  for (i=0; i<CHART_currencyPairCount; i++)
  {
    if (StringFind(CADpairs[i],"CAD",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;               // then run inverted function

    else
      isInvertedPair = false;

    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
            PipChange = CPM_update_cell_inverted (CADpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(CADpairs[i], periodValues[k], barsBack, x_position,
y_position);

          CADStrengths[k] = CPM_addToStrength(CADStrengths[k], PipChange);
        }

        else CPM_alertCondition(CADpairs[i], isInvertedPair, x_position, y_position);

        y_position++;

      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(CADStrengths[k], x_position, y_position);
          y_position++;
        }
```

```
        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }

    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }


////////////////////////////////////////////////////////////////  GBP

  // NEW CHART
  ChartCount++;
    if (ChartCount%NewLineValue == 0)
      x_position = 0;
    else
      x_position++;

  y_position = ChartCount/NewLineValue* CHART_periodCount;

  for (k=0; k<CHART_periodCount-1; k++)
    GBPStrengths[k] = 0;

  for (i=0; i<CHART_currencyPairCount; i++)
  {
    if (StringFind(GBPpairs[i],"GBP",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;               // then run inverted function

    else
      isInvertedPair = false;

    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
            PipChange = CPM_update_cell_inverted (GBPpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(GBPpairs[i], periodValues[k], barsBack, x_position,
y_position);

          GBPStrengths[k] = CPM_addToStrength(GBPStrengths[k], PipChange);
        }

        else CPM_alertCondition(GBPpairs[i], isInvertedPair, x_position, y_position);

        y_position++;
```

```
      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(GBPStrengths[k], x_position, y_position);
          y_position++;
        }

        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }

    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }


////////////////////////////////////////////////////////////////  CHF

  // NEW CHART
  ChartCount++;
    if (ChartCount%NewLineValue == 0)
      x_position = 0;
    else
      x_position++;

  y_position = ChartCount/NewLineValue* CHART_periodCount;

  for (k=0; k<CHART_periodCount-1; k++)
    CHFStrengths[k] = 0;

  for (i=0; i<CHART_currencyPairCount; i++)
  {
    if (StringFind(CHFpairs[i],"CHF",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;               // then run inverted function

    else
      isInvertedPair = false;


    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
```

```
            PipChange = CPM_update_cell_inverted (CHFpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(CHFpairs[i], periodValues[k], barsBack, x_position,
y_position);

          CHFStrengths[k] = CPM_addToStrength(CHFStrengths[k], PipChange);
        }

        else CPM_alertCondition(CHFpairs[i], isInvertedPair, x_position, y_position);

        y_position++;

      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(CHFStrengths[k], x_position, y_position);
          y_position++;
        }

        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }


    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }


////////////////////////////////////////////////////////////////  NZD

  // NEW CHART
  ChartCount++;
    if (ChartCount%NewLineValue == 0)
      x_position = 0;
    else
      x_position++;

  y_position = ChartCount/NewLineValue* CHART_periodCount;

  for (k=0; k<CHART_periodCount-1; k++)
    NZDStrengths[k] = 0;

  for (i=0; i<CHART_currencyPairCount; i++)
  {
```

```
    if (StringFind(NZDpairs[i],"NZD",0) > 0) // if the currency is not at beginning of pair
      isInvertedPair = true;               // then run inverted function

    else
      isInvertedPair = false;


    if ( i< CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          if (isInvertedPair)
            PipChange = CPM_update_cell_inverted (NZDpairs[i], periodValues[k], barsBack, x_position,
y_position);

          else
            PipChange = CPM_update_cell(NZDpairs[i], periodValues[k], barsBack, x_position,
y_position);

          NZDStrengths[k] = CPM_addToStrength(NZDStrengths[k], PipChange);
        }

        else CPM_alertCondition(NZDpairs[i], isInvertedPair, x_position, y_position);

        y_position++;

      }
    }

    if ( i == CHART_currencyPairCount-1)
    {
      for (k=0; k<CHART_periodCount; k++)
      {
        if( k < CHART_periodCount-1)
        {
          CPM_printStrength(NZDStrengths[k], x_position, y_position);
          y_position++;
        }

        else ObjectDelete("PairValue"+ (x_position*100) + y_position);
      }
    }


    if (i != CHART_currencyPairCount-1)
    {
      x_position++;
      y_position = ChartCount/NewLineValue* CHART_periodCount;
    }
  }

}
```

```
//+------------------------------------------------------------------+
//| Custom indicator update period functions                         |
//+------------------------------------------------------------------+
double CPM_update_cell (string cPair,int period, int currentBar, int x, int y)
{
   double delta_pips = CPM_calcPIPS(cPair, period, 1);

   if(delta_pips >= 0)
     ObjectSetText("PairValue"+ (x*100) + y, CPM_dec_align(delta_pips),fontSize,"Courier
New",DarkOliveGreen);

   else
      ObjectSetText("PairValue"+ (x*100) + y, CPM_dec_align(delta_pips),fontSize,"Courier
New",FireBrick);

   return (delta_pips );
}

//+------------------------------------------------------------------------+
//| Custom indicator update period functions inverted (EURUSD -> USDEUR)  |
//+------------------------------------------------------------------------+

double CPM_update_cell_inverted(string cPair, int period, int currentBar, int x, int y)
{
   double delta_pips = CPM_calcPIPS(cPair, period, -1);


   if(delta_pips >= 0)
      ObjectSetText("PairValue"+ (x*100) + y, CPM_dec_align(delta_pips),fontSize,"Courier
New",DarkOliveGreen);


   else
      ObjectSetText("PairValue"+ (x*100) + y, CPM_dec_align(delta_pips),fontSize,"Courier
New",FireBrick);

   return (delta_pips);
 }

//+------------------------------------------------------------------------+
//| Custom indicator update period functions inverted (EURUSD -> USDEUR)  |
//+------------------------------------------------------------------------+

double CPM_printStrength(double strength, int x, int y)
{

   if(strength >= 0)
      ObjectSetText("PairValue"+ (x*100) + y, CPM_dec_align(strength),fontSize,"Courier
New",DarkOliveGreen);


   else
      ObjectSetText("PairValue"+ (x*100) + y, CPM_dec_align(strength),fontSize,"Courier New",FireBrick);

   return (strength);
 }
```

```
//+------------------------------------------------------------------+
//| Custom indicator CPM_addToStrength function                      |
//+------------------------------------------------------------------+
/*
   Adds pip changes across certain timeframe to evaluate the strenght of currency
*/
double CPM_addToStrength(double existing, double pipchange )
{
   return (existing + pipchange );

}




//+------------------------------------------------------------------+
//| Dynamic PIP Overview CPM_dec_align helper function               |
//+------------------------------------------------------------------+
/* converts double to string,
 * adds spaces in front of string to align decimal
 * returns string to be printed.
 */
string CPM_dec_align(double pips)
{

   string pipstring = DoubleToStr(MathAbs(pips), 1);
   int count = 4 - StringFind(pipstring,".",0);

   string aligned;

   while (count > 0)
   {
      aligned = aligned + " ";
      count--;
   }

   aligned = aligned + pipstring;

   return (aligned);
}



//+------------------------------------------------------------------+
//| Currency-Period Matrix CPM_calcPIPS function                     |
//+------------------------------------------------------------------+
/* updatePair takes a curreny pair and a period and calculates pip change
      calculates change in pips for each column for the given row
*/
double CPM_calcPIPS(string cPair, int period, int invert_muliplier)
{
   int bars_included;    // Bars being included in periodicity
   double period_open;   // Opening price of the timeframe
   double period_close;  // Closing price of the timeframe
```

```
//determines pip decimal for the currency pair
  double PIPmultiplier = 10000.0;                // default multiplier to display pips.
  if (StringFind(cPair,"JPY",0) >= 0) // if the currency pair compares to JPY...
     PIPmultiplier = 100.0;                      // ...changes multiplier to represent pips for JPY


  switch(period)
  {

     case PERIOD_M5:
        // update 5min using past 5 bars in M1 periodicity.
        bars_included = 5;
        period_open   = PIPmultiplier * iClose(cPair,PERIOD_M1, barsBack + bars_included);
        period_close  = PIPmultiplier * iClose(cPair,PERIOD_M1, barsBack);
        return ((invert_muliplier)*(period_close - period_open));

      //------------------------------------------------------------------------------

     case PERIOD_M15:
        // update 15min using past 15 bars in M1 periodicity.
        bars_included = 15;
        period_open   = PIPmultiplier * iClose(cPair,PERIOD_M1, barsBack + bars_included);
        period_close  = PIPmultiplier * iClose(cPair,PERIOD_M1, barsBack);
        return ((invert_muliplier)*(period_close - period_open));

        //------------------------------------------------------------------------------

     case PERIOD_M30:
        // update 30min array position using past 6 bars in M5 periodicity.
        bars_included = 6;
        period_open   = PIPmultiplier * iClose(cPair,PERIOD_M5, barsBack + bars_included);
        period_close  = PIPmultiplier * iClose(cPair,PERIOD_M5, barsBack);
        return ((invert_muliplier)*(period_close - period_open));

        //------------------------------------------------------------------------------

     case PERIOD_H1:
        // update 1hr array position using past 12 bars in M5 periodicity.
        bars_included = 12;
        period_open   = PIPmultiplier * iClose(cPair,PERIOD_M5, barsBack + bars_included);
        period_close  = PIPmultiplier * iClose(cPair,PERIOD_M5, barsBack);
        return ((invert_muliplier)*(period_close - period_open));

        //------------------------------------------------------------------------------

     case PERIOD_H4:
        // update 4hr array position using past 8 bars in M30 periodicity.
        bars_included = 8;
        period_open   = PIPmultiplier * iClose(cPair,PERIOD_M30, barsBack + bars_included);
        period_close  = PIPmultiplier * iClose(cPair,PERIOD_M30, barsBack);
        return ((invert_muliplier)*(period_close - period_open));

        //------------------------------------------------------------------------------

     case PERIOD_D1:
        // update daily array position using past 6 bars in H4 periodicity.
```

```
      bars_included = 6;
      period_open   = PIPmultiplier * iClose(cPair,PERIOD_H4, barsBack + bars_included);
      period_close  = PIPmultiplier * iClose(cPair,PERIOD_H4, barsBack);
      return ((invert_muliplier)*(period_close - period_open));

      //------------------------------------------------------------------------------

    case PERIOD_W1:
      // update weekly array position using past 7 bars in D1 periodicity.
      bars_included = 7;
      period_open   = PIPmultiplier * iClose(cPair,PERIOD_D1, barsBack + bars_included);
      period_close  = PIPmultiplier * iClose(cPair,PERIOD_D1, barsBack);
      return ((invert_muliplier)*(period_close - period_open));

      //------------------------------------------------------------------------------

    case PERIOD_MN1:
      // update month array position using past 4 bars in W1 periodicity.
      bars_included = 4;
      period_open   = PIPmultiplier * iClose(cPair,PERIOD_W1, barsBack + bars_included);
      period_close  = PIPmultiplier * iClose(cPair,PERIOD_W1, barsBack);
      return ((invert_muliplier)*(period_close - period_open));

    default:
      return (0);
  }
}
```

```
//+------------------------------------------------------------------+
//| Global MA Analysis CPM_alertCondition function                   |
//+------------------------------------------------------------------+
//RUNS CONDITIONS AND RETURNS THE TRADE COMMAND OF "BUY", "SELL", "CLOSE
POSITION", OR "WAIT"
void CPM_alertCondition(string  currencyPair, bool inverted, int x, int y){

  //determines pip decimal for the currency pair
  double PIPdecimal = 10000.0;              // default multiplier to display pips.
  if (StringFind(currencyPair,"JPY",0) >= 0) // if the currency pair compares to JPY...
    PIPdecimal = 100.0;                     // ...changes multiplier to represent pips for JPY

  double buy_cross_buffer_pips = BUY_CROSS_BUFFER / PIPdecimal; // converts the
CROSS_BUFFER int to the pips decimal
  double sell_cross_buffer_pips = SELL_CROSS_BUFFER / PIPdecimal; // converts the
CROSS_BUFFER int to the pips decimal
```

```
  double close_cross_buffer_pips = CLOSE_CROSS_BUFFER / PIPdecimal; // converts the
CROSS_BUFFER int to the pips decimal

  int shortEMA_period = EMA1;
  int longEMA_period  = EMA1;

  //determines which EMA is larger
  if (EMA1 < EMA2)
    longEMA_period  = EMA2;
  else
    shortEMA_period = EMA2;

  // calculates exponential moving average for the two EMA timeframes for current bar.
  double shortEMA_current = iMA(currencyPair, Period(), shortEMA_period, 0, MODE_EMA, 0,
HISTORY_SHIFT);
  double longEMA_current  = iMA(currencyPair, Period(), longEMA_period , 0, MODE_EMA, 0,
HISTORY_SHIFT);

  // calculates exponential moving average for the two EMA timeframes for previous bar
  double shortEMA_previous = iMA(currencyPair, Period(), shortEMA_period, 0, MODE_EMA, 0,
HISTORY_SHIFT+1);
  double longEMA_previous  = iMA(currencyPair, Period(), longEMA_period , 0, MODE_EMA, 0,
HISTORY_SHIFT+1);

  // calculates simple moving average for SMA and closing price of the bar for later comparison
  double SMA_current    = iMA(currencyPair, Period(), SMA, 0, MODE_SMA, 0, HISTORY_SHIFT);
  double SMA_previous   = iMA(currencyPair, Period(), SMA, 0, MODE_SMA, 0, HISTORY_SHIFT+1);

  double price_current  = iClose(currencyPair, Period(), HISTORY_SHIFT+1);
  double price_previous = iClose(currencyPair, Period(), HISTORY_SHIFT+1);

  // BUY CONDITION 1
  // - price crosses above SMA (including cross_buffer safe-guard)
  // - shortEMA is above longEMA (including cross_buffer safe-guard)
  if (price_current > SMA_current + buy_cross_buffer_pips &&
      price_previous < SMA_previous &&
        shortEMA_current > longEMA_current + buy_cross_buffer_pips)
        {
          if (inverted)
            ObjectSetText("PairValue"+ (x*100) + y, " sell",fontSize,"Courier New",IndianRed);
          else
            ObjectSetText("PairValue"+ (x*100) + y, "  buy",fontSize,"Courier New",OliveDrab);
        }

  // BUY CONDITION 2
  // - shortEMA crosses above longEMA (including cross_buffer safe-guard)
  // - price is above SMA (including cross_buffer safe-guard)
  if (shortEMA_current > longEMA_current + buy_cross_buffer_pips &&
      shortEMA_previous < longEMA_previous &&
        price_current > SMA_current + buy_cross_buffer_pips)
        {
          if (inverted)
            ObjectSetText("PairValue"+ (x*100) + y, " sell",fontSize,"Courier New",IndianRed);
          else
            ObjectSetText("PairValue"+ (x*100) + y, "  buy",fontSize,"Courier New",OliveDrab);
        }
```

```
// SELL CONDIITON 1
// - price crosses under SMA (including cross_buffer safe-guard)
// - shortEMA is below longEMA (including cross_buffer safe-guard)
if (price_current < SMA_current - sell_cross_buffer_pips &&
     price_previous > SMA_previous &&
       shortEMA_current < longEMA_current - sell_cross_buffer_pips)
       {
         if (inverted)
           ObjectSetText("PairValue"+ (x*100) + y, "  buy",fontSize,"Courier New",OliveDrab);
         else
           ObjectSetText("PairValue"+ (x*100) + y, " sell",fontSize,"Courier New",IndianRed);
       }

// SELL CONDITION 2
// - shortEMA crosses under longEMA (including cross_buffer safe-guard)
// - price is below SMA (including cross_buffer safe-guard)
if (shortEMA_current < longEMA_current - sell_cross_buffer_pips &&
     shortEMA_previous > longEMA_previous &&
       price_current < SMA_current - sell_cross_buffer_pips)
       {
         if (inverted)
           ObjectSetText("PairValue"+ (x*100) + y, "  buy",fontSize,"Courier New",OliveDrab);
         else
           ObjectSetText("PairValue"+ (x*100) + y, " sell",fontSize,"Courier New",IndianRed);
       }

// CLOSE POSITION CONDITION 1
// - shortEMA crosses above longEMA (including cross_buffer safe-guard)
// - price is below SMA50 (including cross_buffer safe-guard)
if (shortEMA_current > longEMA_current  + close_cross_buffer_pips&&
     shortEMA_previous < longEMA_previous &&
       price_current < SMA_current + close_cross_buffer_pips)
       {
         ObjectSetText("PairValue"+ (x*100) + y, "close",fontSize,"Courier New",MediumPurple);
       }

// CLOSE POSITION CONDITION 2
// - shortEMA crosses under longEMA (including cross_buffer safe-guard)
// - price is above SMA50 (including cross_buffer safe-guard)
if (shortEMA_current < longEMA_current  - close_cross_buffer_pips&&
     shortEMA_previous > longEMA_previous &&
       price_current > SMA_current - close_cross_buffer_pips)
       {
         ObjectSetText("PairValue"+ (x*100) + y, "close",fontSize,"Courier New",MediumPurple);
       }


// CLOSE POSITION CONDITION 3
// - price crosses under SMA (including cross_buffer safe-guard)
// - shortEMA is above longEMA (including cross_buffer safe-guard)
if (price_current < SMA_current - close_cross_buffer_pips &&
     price_previous > SMA_previous &&
       shortEMA_current > longEMA_current + close_cross_buffer_pips)
       {
         ObjectSetText("PairValue"+ (x*100) + y, "close",fontSize,"Courier New",MediumPurple);
```

```
        }

  // CLOSE POSITION CONDITION 4
  // - price crosses above SMA (including cross_buffer safe-guard)
  // - shortEMA is below longEMA (including cross_buffer safe-guard)
  if (price_current > SMA_current + buy_cross_buffer_pips &&
      price_previous < SMA_previous &&
        shortEMA_current > longEMA_current + buy_cross_buffer_pips)
        {
          ObjectSetText("PairValue"+ (x*100) + y, "close",fontSize,"Courier New",MediumPurple);
        }

  else ObjectSetText("PairValue"+ (x*100) + y, "  n/a",fontSize,"Courier New",DarkSlateGray);
}


/**************************************************************************************************
--------------------------------------------------------------------------------------------
END                              CHART FUNCTIONS                                  END
--------------------------------------------------------------------------------------------
**************************************************************************************************/









/**************************************************************************************************
--------------------------------------------------------------------------------------------
START                            STRENGTH FUNCTIONS                               START
--------------------------------------------------------------------------------------------
**************************************************************************************************/

// updates ALL array positions for currency strengths
void fillStrength(){
  for (int i = Strengths_Displayed; i > 0; i--)
    update_PipsAndPositives(i);
}
// any additional helper functions go here


/**************************************************************************************************
--------------------------------------------------------------------------------------------
START                        UPDATE STRENGTH PARAMETER FUNCTIONS                        START
--------------------------------------------------------------------------------------------
**************************************************************************************************/

// updates the pipsum[] and positivescount[] for the most recent bar of each currency
// on a new bar, rather than recalculating other strengths, push older bars over one in the array
void update_PipsAndPositives(int bar){
```

```
//*******************************
// array_push function here
//*******************************
if (iVolume(Symbol(),Period(),0) < tickvolume)
  array_push();

tickvolume = iVolume(Symbol(),Period(),0);

double PIPchange;
int Cpos = 0;

//------------------------------------------------------------------ USD
USDpipsum[bar] = 0;
USDpositivescount[bar] = 0;

for (int Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
  // if the pair starts with USD

  if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
    PIPchange = getPIPchange(Ppos, bar);
    USDpipsum[bar] = USDpipsum[bar] + PIPchange;
    if (PIPchange >= 0)
      USDpositivescount[bar] = USDpositivescount[bar] + 1;
  }

  // if the pair ends with USD
  else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
    PIPchange = getPIPchange(Ppos, bar);
    USDpipsum[bar] = USDpipsum[bar] - PIPchange;
    if (PIPchange <= 0)
      USDpositivescount[bar] = USDpositivescount[bar] + 1;
  }
}
 currencypipsum[Cpos] = USDpipsum[bar];
Cpos++;

//------------------------------------------------------------------ EUR
EURpipsum[bar] = 0;
EURpositivescount[bar] = 0;

for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
  // if the pair starts with EUR

  if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
    PIPchange = getPIPchange(Ppos, bar);
    EURpipsum[bar] = EURpipsum[bar] + PIPchange;
    if (PIPchange >= 0)
      EURpositivescount[bar] = EURpositivescount[bar] + 1;
  }

  // if the pair ends with EUR
  else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
    PIPchange = getPIPchange(Ppos, bar);
    EURpipsum[bar] = EURpipsum[bar] - PIPchange;
    if (PIPchange <= 0)
      EURpositivescount[bar] = EURpositivescount[bar] + 1;
```

```
   }
}
currencypipsum[Cpos] = EURpipsum[bar];
Cpos++;

//------------------------------------------------------------------ JPY
JPYpipsum[bar] = 0;
JPYpositivescount[bar] = 0;


for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
   // if the pair starts with JPY

   if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
      PIPchange = getPIPchange(Ppos, bar);
      JPYpipsum[bar] = JPYpipsum[bar] + PIPchange;
      if (PIPchange >= 0)
         JPYpositivescount[bar] = JPYpositivescount[bar] + 1;
   }

   // if the pair ends with JPY
   else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
      PIPchange = getPIPchange(Ppos, bar);
      JPYpipsum[bar] = JPYpipsum[bar] - PIPchange;
      if (PIPchange <= 0)
         JPYpositivescount[bar] = JPYpositivescount[bar] + 1;
   }
}
currencypipsum[Cpos] = JPYpipsum[bar];
Cpos++;

//------------------------------------------------------------------ AUD
AUDpipsum[bar] = 0;
AUDpositivescount[bar] = 0;

for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
   // if the pair starts with AUD

   if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
      PIPchange = getPIPchange(Ppos, bar);
      AUDpipsum[bar] = AUDpipsum[bar] + PIPchange;
      if (PIPchange >= 0)
         AUDpositivescount[bar] = AUDpositivescount[bar] + 1;
   }

   // if the pair ends with AUD
   else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
      PIPchange = getPIPchange(Ppos, bar);
      AUDpipsum[bar] = AUDpipsum[bar] - PIPchange;
      if (PIPchange <= 0)
         AUDpositivescount[bar] = AUDpositivescount[bar] + 1;
   }
}
currencypipsum[Cpos] = AUDpipsum[bar];
Cpos++;
```

```
//------------------------------------------------------------------ CAD

CADpipsum[bar] = 0;
CADpositivescount[bar] = 0;

for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
  // if the pair starts with CAD

  if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
    PIPchange = getPIPchange(Ppos, bar);
    CADpipsum[bar] = CADpipsum[bar] + PIPchange;
    if (PIPchange >= 0)
      CADpositivescount[bar] = CADpositivescount[bar] + 1;
  }

  // if the pair ends with CAD
  else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
    PIPchange = getPIPchange(Ppos, bar);
    CADpipsum[bar] = CADpipsum[bar] - PIPchange;
    if (PIPchange <= 0)
      CADpositivescount[bar] = CADpositivescount[bar] + 1;
  }
}
currencypipsum[Cpos] = CADpipsum[bar];
Cpos++;

//------------------------------------------------------------------ GBP

GBPpipsum[bar] = 0;
GBPpositivescount[bar] = 0;

for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
  // if the pair starts with GBP

  if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
    PIPchange = getPIPchange(Ppos, bar);
    GBPpipsum[bar] = GBPpipsum[bar] + PIPchange;
    if (PIPchange >= 0)
      GBPpositivescount[bar] = GBPpositivescount[bar] + 1;
  }

  // if the pair ends with GBP
  else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
    PIPchange = getPIPchange(Ppos, bar);
    GBPpipsum[bar] = GBPpipsum[bar] - PIPchange;
    if (PIPchange <= 0)
      GBPpositivescount[bar] = GBPpositivescount[bar] + 1;
  }
}
currencypipsum[Cpos] = GBPpipsum[bar];
Cpos++;

//------------------------------------------------------------------ CHF

CHFpipsum[bar] = 0;
```

```
   CHFpositivescount[bar] = 0;

   for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
     // if the pair starts with CHF

     if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
        PIPchange = getPIPchange(Ppos, bar);
        CHFpipsum[bar] = CHFpipsum[bar] + PIPchange;
        if (PIPchange >= 0)
           CHFpositivescount[bar] = CHFpositivescount[bar] + 1;
     }

     // if the pair ends with CHF
     else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
        PIPchange = getPIPchange(Ppos, bar);
        CHFpipsum[bar] = CHFpipsum[bar] - PIPchange;
        if (PIPchange <= 0)
           CHFpositivescount[bar] = CHFpositivescount[bar] + 1;
     }
   }
   currencypipsum[Cpos] = CHFpipsum[bar];
   Cpos++;

   //------------------------------------------------------------------ NZD
   NZDpipsum[bar] = 0;
   NZDpositivescount[bar] = 0;

   for (Ppos = 0; Ppos < Strengths_Displayed; Ppos++) {
     // if the pair starts with NZD

     if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 0){
        PIPchange = getPIPchange(Ppos, bar);
        NZDpipsum[bar] = NZDpipsum[bar] + PIPchange;
        if (PIPchange >= 0)
           NZDpositivescount[bar] = NZDpositivescount[bar] + 1;
     }

     // if the pair ends with NZD
     else if (StringFind(pairnames[Ppos],currencynames[Cpos],0) == 3){
        PIPchange = getPIPchange(Ppos, bar);
        NZDpipsum[bar] = NZDpipsum[bar] - PIPchange;
        if (PIPchange <= 0)
           NZDpositivescount[bar] = NZDpositivescount[bar] + 1;
     }
   }

  currencypipsum[Cpos] = NZDpipsum[bar];

  update_Ranks(bar);
}


// pushes array contents over one, replacing the last position and leaving position 0 open for new bar
void array_push(){

   for (int pos = Strengths_Displayed-1; pos >= 0; pos--){
```

```
      USDstrengths [pos+1] = USDstrengths[pos];
      EURstrengths [pos+1] = EURstrengths[pos];
      JPYstrengths [pos+1] = JPYstrengths[pos];
      AUDstrengths [pos+1] = AUDstrengths[pos];
      CADstrengths [pos+1] = CADstrengths[pos];
      GBPstrengths [pos+1] = GBPstrengths[pos];
      CHFstrengths [pos+1] = CHFstrengths[pos];
      NZDstrengths [pos+1] = NZDstrengths[pos];

      USDpipsum [pos+1] = USDpipsum[pos];
      EURpipsum [pos+1] = EURpipsum[pos];
      JPYpipsum [pos+1] = JPYpipsum[pos];
      AUDpipsum [pos+1] = AUDpipsum[pos];
      CADpipsum [pos+1] = CADpipsum[pos];
      GBPpipsum [pos+1] = GBPpipsum[pos];
      CHFpipsum [pos+1] = CHFpipsum[pos];
      NZDpipsum [pos+1] = NZDpipsum[pos];

      USDranked [pos+1] = USDranked[pos];
      EURranked [pos+1] = EURranked[pos];
      JPYranked [pos+1] = JPYranked[pos];
      AUDranked [pos+1] = AUDranked[pos];
      CADranked [pos+1] = CADranked[pos];
      GBPranked [pos+1] = GBPranked[pos];
      CHFranked [pos+1] = CHFranked[pos];
      NZDranked [pos+1] = NZDranked[pos];

      USDpositivescount [pos+1] = USDpositivescount[pos];
      EURpositivescount [pos+1] = EURpositivescount[pos];
      JPYpositivescount [pos+1] = JPYpositivescount[pos];
      AUDpositivescount [pos+1] = AUDpositivescount[pos];
      CADpositivescount [pos+1] = CADpositivescount[pos];
      GBPpositivescount [pos+1] = GBPpositivescount[pos];
      CHFpositivescount [pos+1] = GBPpositivescount[pos];
      NZDpositivescount [pos+1] = NZDpositivescount[pos];
   }
}



// returns the pip change of the currency pair
double getPIPchange(int pos, int bar){

   int bars_included;
   double period_open;
   double period_close;
   string cPair = pairnames[pos];

   double PIPmultiplier = 10000.0;              // default multiplier to display pips.
   if (StringFind(pairnames[pos],"JPY",0) >= 0) // if the currency pair compares to JPY...
      PIPmultiplier = 100.0;

   /*
   period_open   = PIPmultiplier * iOpen(cPair,Period(), bar);
   period_close  = PIPmultiplier * iClose(cPair,Period(), bar);
```

```
 return (period_close - period_open);
*/

 switch(Period())
 {
   case PERIOD_M1:
     // update 5min using past 5 bars in M1 periodicity.
     bars_included = 1;
     period_open   = PIPmultiplier * iClose(cPair,PERIOD_M1, bar*bars_included + bars_included);
     period_close  = PIPmultiplier * iClose(cPair,PERIOD_M1, bar*bars_included);
     return (period_close - period_open);

    //------------------------------------------------------------------------------

   case PERIOD_M5:
     // update 5min using past 5 bars in M1 periodicity.
     bars_included = 5;
     period_open   = PIPmultiplier * iClose(cPair,PERIOD_M1, bar*bars_included + bars_included);
     period_close  = PIPmultiplier * iClose(cPair,PERIOD_M1, bar*bars_included);
     return (period_close - period_open);

    //------------------------------------------------------------------------------

   case PERIOD_M15:
     // update 15min using past 15 bars in M1 periodicity.
     bars_included = 15;
     period_open   = PIPmultiplier * iClose(cPair,PERIOD_M1, bar*bars_included + bars_included);
     period_close  = PIPmultiplier * iClose(cPair,PERIOD_M1, bar*bars_included);
     return (period_close - period_open);

     //------------------------------------------------------------------------------

   case PERIOD_M30:
     // update 30min array position using past 6 bars in M5 periodicity.
     bars_included = 6;
     period_open   = PIPmultiplier * iClose(cPair,PERIOD_M5, bar*bars_included + bars_included);
     period_close  = PIPmultiplier * iClose(cPair,PERIOD_M5, bar*bars_included);
     return (period_close - period_open);

     //------------------------------------------------------------------------------

   case PERIOD_H1:
     // update 1hr array position using past 12 bars in M5 periodicity.
     bars_included = 12;
     period_open   = PIPmultiplier * iClose(cPair,PERIOD_M5, bar*bars_included + bars_included);
     period_close  = PIPmultiplier * iClose(cPair,PERIOD_M5, bar*bars_included);
     return (period_close - period_open);

     //------------------------------------------------------------------------------

   case PERIOD_H4:
     // update 4hr array position using past 8 bars in M30 periodicity.
     bars_included = 8;
     period_open   = PIPmultiplier * iClose(cPair,PERIOD_M30,  bar*bars_included + bars_included);
     period_close  = PIPmultiplier * iClose(cPair,PERIOD_M30, bar*bars_included);
     return (period_close - period_open);
```

```
      //-------------------------------------------------------------------------

    case PERIOD_D1:
      // update daily array position using past 6 bars in H4 periodicity.
      bars_included = 6;
      period_open   = PIPmultiplier * iClose(cPair,PERIOD_H4, bar*bars_included + bars_included);
      period_close  = PIPmultiplier * iClose(cPair,PERIOD_H4, bar*bars_included);
      return (period_close - period_open);

      //-------------------------------------------------------------------------

    case PERIOD_W1:
      // update weekly array position using past 7 bars in D1 periodicity.
      bars_included = 7;
      period_open   = PIPmultiplier * iClose(cPair,PERIOD_D1, bar*bars_included + bars_included);
      period_close  = PIPmultiplier * iClose(cPair,PERIOD_D1, bar*bars_included);
      return (period_close - period_open);

      //-------------------------------------------------------------------------

    case PERIOD_MN1:
      // update month array position using past 4 bars in W1 periodicity.
      bars_included = 4;
      period_open   = PIPmultiplier * iClose(cPair,PERIOD_W1, bar*bars_included + bars_included);
      period_close  = PIPmultiplier * iClose(cPair,PERIOD_W1, bar*bars_included);
      return (period_close - period_open);

    default:
      return (0);
  }
}



/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
END                        UPDATE STRENGTH PARAMETER FUNCTIONS                        END
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/









/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                      UPDATE RANKED STRENGTH FUNCTIONS                      START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/

// updates and assigns the rankednames[] rankedpipsum[], and ranked arrays of each currency
```

```
void update_Ranks(int bar){


  // Applies updated values to rank arrays.
  for (int rank_pos = 0; rank_pos < STR_CURRENCIES; rank_pos++)
  {
    int match_found = 0; // nested for loop break for efficiency

    for (int curr_pos = 0; curr_pos < STR_CURRENCIES && match_found == 0; curr_pos++)
    {
      if (currencynames[curr_pos] == rankednames[rank_pos])
      {
        rankedpipsum[rank_pos] = currencypipsum[curr_pos];
        match_found = 1;                    // ends nested for loop.
      }
    }
  }



  //sorts the rank arrays in ascending order.
  int inorder = 0; // initially assumes the arrays are not ordered.
  for (int round = STR_CURRENCIES-1; round > 0 && inorder == 0; round--)
  {
    inorder = 1;        // assumes pairs are in ranked order for the round
    rank_pos = 0;       // resets array traversing counter

    while (rank_pos < round)
    {
      if (rankedpipsum[rank_pos] <= rankedpipsum[rank_pos+1])
        rank_pos++;          // rank arrays are in order

      else
      {
        inorder = 0;         // rank arrays are not in order.

        //swaps the positions
        string temp_name = rankednames[rank_pos];
        int temp_PIPS = rankedpipsum [rank_pos];

        rankednames[rank_pos] = rankednames[rank_pos + 1];
        rankedpipsum[rank_pos] = rankedpipsum [rank_pos + 1];

        rankednames[rank_pos + 1] = temp_name;
        rankedpipsum [rank_pos + 1] = temp_PIPS;
      }
    }
  }



  //matches new rank order to currency pairs
  for (curr_pos = 0; curr_pos < STR_CURRENCIES; curr_pos++)
  {
    match_found = 0; // nested for loop break for efficiency
```

```
    for (rank_pos = 0; rank_pos < STR_CURRENCIES && match_found == 0; rank_pos++)
    {
      if (currencynames[curr_pos] == rankednames[rank_pos])
      {
        currencyrank[curr_pos] = rank_pos + 1; // applies the rank number to the currency pair.
        match_found = 1;                  // ends nested for loop.
      }
    }
  }

  applyranks(bar);
  update_Strength(bar);
}


void applyranks(int bar){
  int Cpos = 0;
  USDranked [bar] = currencyrank[Cpos];
  Cpos++;
  EURranked [bar] = currencyrank[Cpos];
  Cpos++;
  JPYranked [bar] = currencyrank[Cpos];
  Cpos++;
  AUDranked [bar] = currencyrank[Cpos];
  Cpos++;
  CADranked [bar] = currencyrank[Cpos];
  Cpos++;
  GBPranked [bar] = currencyrank[Cpos];
  Cpos++;
  CHFranked [bar] = currencyrank[Cpos];
  Cpos++;
  NZDranked [bar] = currencyrank[Cpos];
}

// include any helper functions for update_Ranks here



/*************************************************************************************************
-------------------------------------------------------------------------------------------------
END                        UPDATE RANKED STRENGTH FUNCTIONS                        END
-------------------------------------------------------------------------------------------------
*************************************************************************************************/




/*************************************************************************************************
-------------------------------------------------------------------------------------------------
START                        UPDATE STRENGTH FUNCTIONS                        START
-------------------------------------------------------------------------------------------------
*************************************************************************************************/

// updates  the currency strength array for each currency
```

```
void update_Strength(int bar){
   USDstrengths [bar] = calculateStrength(USDranked[bar], USDpositivescount[bar]);
   EURstrengths [bar] = calculateStrength(EURranked[bar], EURpositivescount[bar]);
   JPYstrengths [bar] = calculateStrength(JPYranked[bar], JPYpositivescount[bar]);
   AUDstrengths [bar] = calculateStrength(AUDranked[bar], AUDpositivescount[bar]);
   CADstrengths [bar] = calculateStrength(CADranked[bar], CADpositivescount[bar]);
   GBPstrengths [bar] = calculateStrength(GBPranked[bar], GBPpositivescount[bar]);
   CHFstrengths [bar] = calculateStrength(CHFranked[bar], CHFpositivescount[bar]);
   NZDstrengths [bar] = calculateStrength(NZDranked[bar], NZDpositivescount[bar]);

   print_Update();
}

//determines the strength on a scale of 1-8 based on positive pairs and currency rank
int calculateStrength(int rank, int positive_pairs){
   int strength;

   strength = (rank*rank_weight + ((positive_pairs+1)*positives_weight))/(positives_weight + rank_weight);

   return (strength);
}

// include any helper functions for update_Ranks here



/*************************************************************************************************
-------------------------------------------------------------------------------------------------
END                          UPDATE STRENGTH FUNCTIONS                          END
-------------------------------------------------------------------------------------------------
*************************************************************************************************/




/*************************************************************************************************
-------------------------------------------------------------------------------------------------
START                          PRINT STRENGTH FUNCTIONS                          START
-------------------------------------------------------------------------------------------------
*************************************************************************************************/

// prints the updated strength arrays to the terminal
void print_Update(){

   int title = 0;
   int str = 1;
   for (int pos = Strengths_Displayed-1; pos >= 0; pos--){
      printStrength(currencynames[title] + str, USDstrengths[pos]);
      str++;
   }

   title++;
   str = 1;
   for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
```

```
    printStrength(currencynames[title] + str, EURstrengths[pos]);
    str++;
  }

  title++;
  str = 1;
  for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
    printStrength(currencynames[title] + str, JPYstrengths[pos]);
    str++;
  }

  title++;
  str = 1;
  for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
    printStrength(currencynames[title] + str, AUDstrengths[pos]);
    str++;
  }

  title++;
  str = 1;
  for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
    printStrength(currencynames[title] + str, CADstrengths[pos]);
    str++;
  }

  title++;
  str = 1;
  for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
    printStrength(currencynames[title] + str, GBPstrengths[pos]);
    str++;
  }

  title++;
  str = 1;
  for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
    printStrength(currencynames[title] + str, CHFstrengths[pos]);
    str++;
  }

  title++;
  str = 1;
  for ( pos = Strengths_Displayed-1; pos >= 0; pos--){
    printStrength(currencynames[title] + str, NZDstrengths[pos]);
    str++;
  }
}



void printStrength(string name, int strength){
  switch(strength)
  {
    case 1:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number1);
      break;
    case 2:
```

```
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number2);
      break;
    case 3:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number3);
      break;
    case 4:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number4);
      break;
    case 5:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number5);
      break;
    case 6:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number6);
      break;
    case 7:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number7);
      break;
    case 8:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number8);
      break;
    default:
      ObjectSetText(name, "" + strength ,numberSize-2, monospace, Number8);
     // ObjectSetText(name, CharToStr(110),fontSize,"WingDings",TitleColor);
      break;
  }
}

// include any helper functions for update_Ranks here


/*********************************************************************************************
---------------------------------------------------------------------------------------------
END                          PRINT STRENGTH FUNCTIONS                           END
---------------------------------------------------------------------------------------------
*********************************************************************************************/




/*********************************************************************************************
---------------------------------------------------------------------------------------------
START                        GMAA HELPER/BUILDER FUNCTIONS                      START
---------------------------------------------------------------------------------------------
*********************************************************************************************/

//+------------------------------------------------------------------+
//| Global MA Analysis buildTrackCurrency function                   |
//+------------------------------------------------------------------+
// applies currency tracking settings into an array to be traversed
void buildTrackCurrency()
{
  trackCurrency[1]  = Track_EURUSD;
  trackCurrency[2]  = Track_GBPUSD;
```

```
   trackCurrency[3]  = Track_USDCHF;
   trackCurrency[4]  = Track_USDJPY;
   trackCurrency[5]  = Track_AUDUSD;
   trackCurrency[6]  = Track_EURCHF;
   trackCurrency[7]  = Track_EURGBP;
   trackCurrency[8]  = Track_NZDUSD;
   trackCurrency[9]  = Track_USDCAD;
   trackCurrency[10] = Track_AUDCAD;
   trackCurrency[11] = Track_AUDCHF;
   trackCurrency[12] = Track_CADJPY;
   trackCurrency[13] = Track_CHFJPY;
   trackCurrency[14] = Track_EURAUD;
   trackCurrency[15] = Track_EURCAD;
   trackCurrency[16] = Track_EURNZD;
   trackCurrency[17] = Track_GBPAUD;
   trackCurrency[18] = Track_GBPCAD;
   trackCurrency[19] = Track_GBPCHF;
   trackCurrency[20] = Track_GBPNZD;
   trackCurrency[21] = Track_CADCHF;
   trackCurrency[22] = Track_NZDCHF;
   trackCurrency[23] = Track_NZDCAD;
   trackCurrency[24] = Track_NZDJPY;
   trackCurrency[25] = Track_AUDNZD;
   trackCurrency[26] = Track_AUDJPY;
   trackCurrency[27] = Track_EURJPY;
   trackCurrency[28] = Track_GBPJPY;
}

//+------------------------------------------------------------------+
//| Global MA Analysis isWindowOpen function                         |
//+------------------------------------------------------------------+
// returns integer indicating if the current position is in the timewindow allowed
int isWindowOpen(int pos)
{
   // calculate the time of the current bar
   datetime current_time = iTime(pairnames[pos],Period(),HISTORY_SHIFT);

   //Check Sunday
   if (TimeDayOfWeek(current_time) == 0 &&                    // If it is Sunday
      Trade_on_Sunday == 1 &&                                 // and trading is allowed in Sunday
      isInTime(current_time, Sunday_Open, Sunday_Close))      // and the trading window for Sunday
is Open
         return (1);                                          // return that you are within window

   //Check Monday
   else if (TimeDayOfWeek(current_time) == 1 &&               // If it is Monday
          Trade_on_Monday == 1 &&                            // and trading is allowed in Monday
          isInTime(current_time, Monday_Open, Monday_Close))  // and the trading window for Monday
is Open
             return (1);                                      // return that you are within window


   //Check Tuesday
   else if (TimeDayOfWeek(current_time) == 2 &&               // If it is Tuesday
          Trade_on_Tuesday == 1 &&                           // and trading is allowed in Tuesday
```

```
        isInTime(current_time, Tuesday_Open, Tuesday_Close))     // and the trading window for Tuesday
is Open
          return (1);                                  // return that you are within window


  //Check Wednesday
  else if (TimeDayOfWeek(current_time) == 3 &&                  // If it is Wednesday
        Trade_on_Wednesday == 1 &&                          // and trading is allowed in Wednesday
        isInTime(current_time, Wednesday_Open, Wednesday_Close)) // and the trading window for
Wednesday is Open
          return (1);                                  // return that you are within window


  //Check Thursday
  else if (TimeDayOfWeek(current_time) == 4 &&                  // If it is Thursday
        Trade_on_Thursday == 1 &&                           // and trading is allowed in Thursday
        isInTime(current_time, Thursday_Open, Thursday_Close))   // and the trading window for
Thursday is Open
          return (1);                                  // return that you are within window


  //Check Friday
  else if (TimeDayOfWeek(current_time) == 5 &&                  // If it is Friday
        Trade_on_Friday == 1 &&                             // and trading is allowed in Friday
        isInTime(current_time, Friday_Open, Friday_Close))       // and the trading window for Friday is
Open
          return (1);                                  // return that you are within window


  //Check Saturday
  else if (TimeDayOfWeek(current_time) == 5 &&                  // If it is Saturday
        Trade_on_Saturday == 1 &&                           // and trading is allowed in Saturday
        isInTime(current_time, Saturday_Open, Saturday_Close))   // and the trading window for Saturday
is Open
          return (1);                                  // return that you are within window

  else return (0);
}

//+------------------------------------------------------------------+
//| Global MA Analysis isInTime function                             |
//+------------------------------------------------------------------+
// returns integer indicating if the first datetime is within the second and third
int isInTime( datetime current, datetime opened, datetime closed)
{
  //calculates an integer equivelant to current time
  int current_time = TimeHour(current)*10000 +
                TimeMinute(current)*100 +
                  TimeSeconds(current);

  //calculates an integer equivelant to opening time
  int opening_time = TimeHour(opened)*10000 +
                TimeMinute(opened)*100 +
                  TimeSeconds(opened);

  //calculates an integer equivelant to closing time
```

```
   int closing_time = TimeHour(opened)*10000 +
                      TimeMinute(opened)*100 +
                        TimeSeconds(opened);


   if (current_time >= opening_time && // if current time is >= opening
       current_time <= closing_time)   // and <= closing
         return (1);                   // then return true

   return (0);
}



//+------------------------------------------------------------------+
//| Global MA Analysis printableTimeframe function                   |
//+------------------------------------------------------------------+
/* returns a string declaring the time frame of the indicator */
string printableTimeframe()
{

   switch (Period())
   {
      case PERIOD_M1:
         return ("1 Minute");

      case PERIOD_M5:
         return ("5 Minutes");

      case PERIOD_M15:
         return ("15 Minutes");

      case PERIOD_M30:
         return ("30 Minutes");

      case PERIOD_H1:
         return ("1 Hour");

      case PERIOD_H4:
         return ("4 Hours");

      case PERIOD_D1:
         return ("Daily");

      case PERIOD_W1:
         return ("Weekly");

      case PERIOD_MN1:
         return ("Monthly");

      default:
         return ("VOID");
   }
}

//+------------------------------------------------------------------+
```

```
//| Global MA Analysis updateSpread function                        |
//+------------------------------------------------------------------+
/* updates and returns the spread of the given currency position */
double updateSpread(int pos)
{
  string currencypair = pairnames[pos];
  int multiplier = 10000; // default multiplier to display pips.

  if (StringFind(currencypair,"JPY",0) >= 0) // if the currency pair compares to JPY...
    multiplier = 100;                        // ...changes multiplier to represent pips for JPY

  //Buy at ask, sell at bid
  double Spread = multiplier*(MarketInfo(currencypair, MODE_ASK)-
MarketInfo(currencypair,MODE_BID));

  currencySpreads[pos] = Spread;

  return (Spread);
}


/**************************************************************************************************
--------------------------------------------------------------------------------------------------
END                          GMAA HELPER/BUILDER FUNCTIONS                              END
--------------------------------------------------------------------------------------------------
**************************************************************************************************/









/**************************************************************************************************
--------------------------------------------------------------------------------------------------
START                          ERROR CHECKING FUNCTIONS                              START
--------------------------------------------------------------------------------------------------
**************************************************************************************************/



//+------------------------------------------------------------------+
//| Global MA Analysis ErrCheck function                             |
//+------------------------------------------------------------------+
/* Checks for any possible errors in the history and input
  takes an array position to check history
  returns 0 if there are no errors.
  otherwise returns -1
*/
int ErrCheck(int pos)
{
  string   currencyPair      = pairnames[pos];    // CURRENCY
  int      latest_alert_time = alertTime[pos];    // TIME OF MOST RECENT ALERT FOR CURRENCY
  int      latest_alert_type = alertType[pos];    // TYPE OF MOST RECENT ALERT FOR CURRENCY
```

```
   int      latest_open_alert = openAlert[pos];    // TYPE OF MOST RECENT OPENING ALERT
   datetime current_time      = iTime(currencyPair,Period(),HISTORY_SHIFT);   // TIME OF THE
CURRENT BAR
   int      current_volume    = iVolume(currencyPair,Period(),HISTORY_SHIFT); // TICK VOLUME OF THE
CURRENT BAR.


   // inputError
   int extern_errors = CheckExternalInputError();
   if (extern_errors != 0)
      return (-1);

   // history_error
   int history_errors = CheckHistoryError(currencyPair, pos);
   if (history_errors != 0)
      return (1);



   // if there have been no errors
   inputError = 0;
   history_error[pos] = 0;
   return (0);
}

//+------------------------------------------------------------------+
//| Global MA Analysis CheckExternalInputError function              |
//+------------------------------------------------------------------+
/* Checks all external inputs for any invalid entries.
   returns 0 if there are no errors.
   otherwise returns error number
*/
int CheckExternalInputError()
{
   // SMA must be greater than 1
   if (SMA <= 1){
      if (inputError != -1)
         Alert("External Input Error!!!\nSMA must have size greater than 1.");
      inputError = -1;
      return (2);
   }

   // EMAs must be greater than zero
   if (EMA1 <= 0 || EMA2 <=0){
      if (inputError != -1)
         Alert("External Input Error!!!\nEMA1 and EMA2 must have size greater than 0.");
      inputError = -1;
      return (2);
   }

   // EMAs cannot be equal
   if (EMA1 == EMA2){
      if (inputError != -1)
         Alert("External Input Error!!!\nEMA1 and EMA2 cannot be of equal size.");
      inputError = -1;
      return (3);
```

```
}



//** Check that the Graph and position are valid inputs **//

// History_shift must be at least 0
if (HISTORY_SHIFT < 0){
   if (inputError != -1)
      Alert("External Input Error!!!\nHISTORY_SHIFT cannot be less than 0.");
   inputError = -1;
   return (5);
}

// ALERTS_PER_BAR must be at least 0
if (ALERTS_PER_BAR < 0){
   if (inputError != -1)
      Alert("External Input Error!!!\nALERTS_PER_BAR cannot be less than 0.");
   inputError = -1;
   return (6);
}

// Cross buffers must be at least 0
if (BUY_CROSS_BUFFER < 0){
   if (inputError != -1)
      Alert("External Input Error!!!\nBUY_CROSS_BUFFER cannot be less than 0.");
   inputError = -1;
   return (7);
}

if (SELL_CROSS_BUFFER < 0){
   if (inputError != -1)
      Alert("External Input Error!!!\nSELL_CROSS_BUFFER cannot be less than 0.");
   inputError = -1;
   return (8);
}

if (CLOSE_CROSS_BUFFER < 0){
   if (inputError != -1)
      Alert("External Input Error!!!\nCLOSE_CROSS_BUFFER cannot be less than 0.");
   inputError = -1;
   return (9);
}


//INITIALIZE_BARS must be at least 0
if (INITIALIZE_BARS < 0){
   if (inputError != -1)
      Alert("External Input Error!!!\nINITIALIZE_BARS cannot be less than 0.");
   inputError = -1;
   return (10);
}

//Needs to track at least one currency
if (Track_EURUSD == 0 &&
    Track_GBPUSD == 0 &&
```

```
    Track_USDCHF == 0 &&
    Track_USDJPY == 0 &&
    Track_AUDUSD == 0 &&
    Track_EURCHF == 0 &&
    Track_EURGBP == 0 &&
    Track_NZDUSD == 0 &&
    Track_USDCAD == 0 &&
    Track_AUDCAD == 0 &&
    Track_AUDCHF == 0 &&
    Track_CADJPY == 0 &&
    Track_CHFJPY == 0 &&
    Track_EURAUD == 0 &&
    Track_EURCAD == 0 &&
    Track_EURNZD == 0 &&
    Track_GBPAUD == 0 &&
    Track_GBPCAD == 0 &&
    Track_GBPCHF == 0 &&
    Track_GBPNZD == 0 &&
    Track_CADCHF == 0 &&
    Track_NZDCHF == 0 &&
    Track_NZDCAD == 0 &&
    Track_NZDJPY == 0 &&
    Track_AUDNZD == 0 &&
    Track_AUDJPY == 0 &&
    Track_EURJPY == 0 &&
    Track_GBPJPY == 0){
  if (inputError != -1)
     Alert("External Input Error!!!\nNo purrency pairs are being tracked.");
  inputError = -1;
  return (11);
}


// Needs to be able to trade on at least one day
if (Trade_on_Sunday    == 0 &&
    Trade_on_Monday    == 0 &&
    Trade_on_Tuesday   == 0 &&
    Trade_on_Wednesday == 0 &&
    Trade_on_Thursday  == 0 &&
    Trade_on_Friday    == 0 &&
    Trade_on_Saturday  == 0){
  if (inputError != -1)
     Alert("External Input Error!!!\nAll days are closed to trading.");
  inputError = -1;
  return (12);
}


// open time cannot be less than 0
if (Sunday_Open    < D'00:00:00' ||
    Monday_Open    < D'00:00:00' ||
    Tuesday_Open   < D'00:00:00' ||
    Wednesday_Open < D'00:00:00' ||
    Thursday_Open  < D'00:00:00' ||
    Friday_Open    < D'00:00:00' ||
    Saturday_Open  < D'00:00:00'){
```

```
      if (inputError != -1)
         Alert("External Input Error!!!\nOpen Period for the day cannot be less than 00:00:00.");
      inputError = -1;
      return (13);
   }


   // close time cannot be greater than 23:59:59
   if (Sunday_Close    > D'23:59:59' ||
       Monday_Close    > D'23:59:59' ||
       Tuesday_Close   > D'23:59:59' ||
       Wednesday_Close > D'23:59:59' ||
       Thursday_Close  > D'23:59:59' ||
       Friday_Close    > D'23:59:59' ||
       Saturday_Close  > D'23:59:59'){
      if (inputError != -1)
         Alert("External Input Error!!!\nClose Period for the day cannot be greater than 23:59:59.");
      inputError = -1;
      return (14);
   }

   // close time must be greater than open time
   if (Sunday_Open    > Sunday_Close    ||
       Monday_Open    > Monday_Close    ||
       Tuesday_Open   > Tuesday_Close   ||
       Wednesday_Open > Wednesday_Close ||
       Thursday_Open  > Thursday_Close  ||
       Friday_Open    > Friday_Close    ||
       Saturday_Open  > Saturday_Close ){
      if (inputError != -1)
         Alert("External Input Error!!!\nClose time for the day cannot come before the Open time.");
      inputError = -1;
      return (15);
   }

   inputError = 0;
   return (0);
}


//+------------------------------------------------------------------+
//| Global MA Analysis CheckHistoryError function                    |
//+------------------------------------------------------------------+
/* Checks for any possible gaps in the history
   takes an array position to check history
   returns 0 if there are no errors.
   otherwise returns error number
*/
int CheckHistoryError(string cPair, int pos)
{
   int this_bar = iTime(cPair,Period(),HISTORY_SHIFT) / 60;
   int first_bar;    // time of oldest bar calculated
   int diff_hypoth;  // Ideal time difference between bars
   int diff_actual;  // actual difference between bars
   int lower_moe;    // lower margin of error for gaps
   int upper_moe;    // upper margin of error for gaps
```

```
// make sure enough bars are loaded to calculate SMA
if (iClose(cPair,Period(),HISTORY_SHIFT+SMA) == 0){
  if (history_error[pos] != -1)
    Alert("Incomplete History Warning:  " + cPair + "\nUnable to check " + cPair
        + ".\nIs history loaded for symbol?\nOr is SMA too large?\n");
  history_error[pos] = -1;
  return (16);
}

// make sure there are no gaps in the past (SMA) bars
// using time of current bar and time of SMA bars ago
first_bar   = iTime(cPair,Period(),HISTORY_SHIFT+SMA) / 60;
diff_hypoth = SMA*Period();
diff_actual = this_bar - first_bar;
lower_moe = diff_hypoth - diff_hypoth*GAP_MARGIN_OF_ERROR;
upper_moe = diff_hypoth + diff_hypoth*GAP_MARGIN_OF_ERROR;

if (diff_actual < lower_moe || diff_actual > upper_moe){
  if (history_error[pos] != -1)
    Alert("Incomplete History Warning:  " + cPair + "\nUnable to check " + cPair
        + ".\nGap in history detected while checking SMA.\n"
        + "SMA = " + SMA);
  history_error[pos] = -1;
  return (17);
}


// make sure enough bars are loaded to calculate EMA1
if (iClose(cPair,Period(),HISTORY_SHIFT+EMA1) == 0){
  if (history_error[pos] != -1)
    Alert("Incomplete History Warning:  " + cPair + "\nUnable to check " + cPair
        + ".\nIs history loaded for symbol?\nOr is EMA1 too large?\n");
  history_error[pos] = -1;
  return (18);
}

// make sure there are no gaps in the past (EMA1) bars
// using time of current bar and time of SMA bars ago
first_bar   = iTime(cPair,Period(),HISTORY_SHIFT+EMA1) / 60;
diff_hypoth = EMA1*Period();
diff_actual = this_bar - first_bar;
lower_moe = diff_hypoth - diff_hypoth*GAP_MARGIN_OF_ERROR;
upper_moe = diff_hypoth + diff_hypoth*GAP_MARGIN_OF_ERROR;

if (diff_actual < lower_moe || diff_actual > upper_moe){
  if (history_error[pos] != -1)
    Alert("Incomplete History Warning:  " + cPair + "\nUnable to check " + cPair
        + ".\nGap in history detected while checking EMA1.\n"
        + "EMA1 = " + EMA1);
  history_error[pos] = -1;
  return (19);
}


// make sure enough bars are loaded to calculate EMA2
```

```
  if (iClose(cPair,Period(),HISTORY_SHIFT+EMA2) == 0){
    if (history_error[pos] != -1)
      Alert("Incomplete History Warning:  " + cPair + "\nUnable to check " + cPair
          + ".\nIs history loaded for symbol?\nOr is EMA2 too large?\n");
    history_error[pos] = -1;
    return (20);
  }

  // make sure there are no gaps in the past (EMA2) bars
  // using time of current bar and time of SMA bars ago
  first_bar   = iTime(cPair,Period(),HISTORY_SHIFT+EMA2) / 60;
  diff_hypoth = EMA2*Period();
  diff_actual = this_bar - first_bar;
  lower_moe = diff_hypoth - diff_hypoth*GAP_MARGIN_OF_ERROR;
  upper_moe = diff_hypoth + diff_hypoth*GAP_MARGIN_OF_ERROR;

  if (diff_actual < lower_moe || diff_actual > upper_moe){
    if (history_error[pos] != -1)
      Alert("Incomplete History Warning:  " + cPair + "\nUnable to check " + cPair
          + ".\nGap in history detected while checking EMA2.\n"
          + "EMA2 = " + EMA2);
    history_error[pos] = -1;
    return (21);
  }


  history_error[pos] = 0;
  return (0);
}

/**************************************************************************************************
--------------------------------------------------------------------------------------------------
END                          ERROR CHECKING FUNCTIONS                          END
--------------------------------------------------------------------------------------------------
**************************************************************************************************/







/**************************************************************************************************
--------------------------------------------------------------------------------------------------
START                        GMAA RUNNING ALERT FUNCTIONS                        START
--------------------------------------------------------------------------------------------------
**************************************************************************************************/


//+------------------------------------------------------------------+
//| Global MA Analysis alertTheWorld function                        |
//+------------------------------------------------------------------+
//CHECKS ALL CURRENCIES FOR AN ALERT.
void alertTheWorld()
```

```
{
  int pos = 0; // position in arrays :: used to traverse while loop
  int errors = 0;
  while (pos < pairs && errors >= 0 && inputError == 0)
  {
    //checks for any errors before processing
    if (ErrCheck(pos) == 0)
    {
      if (trackCurrency[pos] == 1)   // If the currency is set to be tracked
      {
        int open = isWindowOpen(pos);    // Finds out if the trading window is open
        pairAlert(pos, open);            // check currency pair in pairnames[pos] for an alert.
      }
    }
    else errors++;

    pos++;
  }

  if (inputError != 0)
    Comment("FXCM Multiple Currency Overview has stopped.");

  else if (errors != 0 && inputError == 0)
    Comment("One or more currency pairs have an incomplete history.\n");

  else if (errors == 0)
    Comment("FXCM Multiple Currency Overview is running...");

  if (inputError == 0){
    CPM_updateMatrix();
    update_PipsAndPositives(0);
  }
}

//+------------------------------------------------------------------+
//| Global MA Analysis pairAlert function                            |
//+------------------------------------------------------------------+
// CHECKS THE CURRENCY IN THE POSITION PROVIDED FOR AN ALERT.
// CREATES AN ALERT IF PROMTPED AND THERE IS NOT ONE FOR THE CURRENT BAR.

void pairAlert(int pos, int open)
{
  string   currencyPair      = pairnames[pos];    // CURRENCY
  double   spread            = updateSpread(pos); // SPREAD OF THE CURRENCY
  int      latest_alert_bar  = alertBar[pos];     // BAR TIME OF MOST RECENT ALERT FOR CURRENCY
  int      latest_alert_time = alertTime[pos];    // EXACT TIME OF MOST RECENT ALERT FOR
CURRENCY
  int      latest_alert_type = alertType[pos];    // TYPE OF MOST RECENT ALERT FOR CURRENCY
  int      latest_open_alert = openAlert[pos];    // TYPE OF MOST RECENT OPENING ALERT
  string   trade_status      = alertCondition(currencyPair);                  // BUY-SELL-WAIT order tradeorder
  int      current_volume    = iVolume(currencyPair,Period(),HISTORY_SHIFT); // VOLUME OF THE
CURRENT BAR.
  int      current_bar       = iTime(currencyPair,Period(),HISTORY_SHIFT);   // OPEN TIME OF THE
CURRENT BAR
  int      current_time      = TimeCurrent();
```

```
  string   print_price       = DoubleToStr(iClose(currencyPair,Period(),HISTORY_SHIFT),5); // PRICE OF
THE CURRENT BAR
  if (StringFind(currencyPair,"JPY",0) >= 0) // if the currency pair compares to JPY...
    print_price       = DoubleToStr(iClose(currencyPair,Period(),HISTORY_SHIFT),3);


  if (current_volume < currencyVolumes[pos])   // If the volume has decreased (indicating a new bar)
    alertCount[pos] = 0;               // clear the number of alerts for the bar

  int alerts_this_bar   = alertCount[pos]; // NUMBER OF ALERTS IN THE GIVEN BAR

  currencyVolumes[pos] = current_volume; // update the recorded currency bar's volume.


  // BUY CONDITION IS MET
  if (open == 1 && trade_status == "BUY" &&          // if the market is open and there is an order to buy
     latest_alert_type < 1)                      // and if the BUY order is NOT being repeated...
  {
    if (alerts_this_bar < ALERTS_PER_BAR ||         // ... and if the number of alerts this bar has not
exceeded the limit
        ALERTS_PER_BAR == 0)                      // or if the limit is zero...
    {
      alertTime[pos] = current_time;               // ... then update the most recent alert time
      alertBar[pos] = current_bar;                 // update the most recent alert bar
      alertType[pos] = 1;                         // update the most recent alert type to BUY
      openAlert[pos] = 1;                         // update the most recent open order type
      alertCount[pos]++;                          // increment the number of alerts created in the current bar

      if (DISPLAY_ALERTS)
        Alert(currencyPair + ":    " + trade_status
          + "\n\nTimeframe: "  + printableTimeframe()
          + "\nPrice:   "    + print_price
          + "\nSpread:   "   + DoubleToStr(spread,2)
          + "\nCause:   "    + AlertCausedBy);   // generate a BUY alert.
    }
  }

  // SELL CONDITION IS MET
  else if (open == 1 && trade_status == "SELL" &&    // if the market is open and there is an order to buy
        latest_alert_type > -1)                 // and if the SELL order is NOT being repeated...
  {
    if (alerts_this_bar < ALERTS_PER_BAR ||         // ... and if the number of alerts this bar has not
exceeded the limit
        ALERTS_PER_BAR == 0)                      // or if the limit is zero...
    {
      alertTime[pos] = current_time;               // ... then update the most recent alert time
      alertBar[pos] = current_bar;                 // update the most recent alert bar
      alertType[pos] = -1;                        // update the most recent alert type to SELL
      openAlert[pos] = -1;                        // update the most recent open order type
      alertCount[pos]++;                          // increment the number of alerts created in the current bar

      if (DISPLAY_ALERTS)
        Alert(currencyPair + ":    " + trade_status
          + "\n\nTimeframe: "  + printableTimeframe()
          + "\nPrice:   "    + print_price
```

```
            + "\nSpread:   "   + DoubleToStr(spread,2)
            + "\nCause:   "    + AlertCausedBy);    // generate a SELL alert.
    }
  }

  // CLOSE POSITION CONDITION IS MET
  else if (trade_status == "CLOSE POSITION" &&        // if there is an order to close position
          latest_alert_type != 0)                  // and if the CLOSE POSITION order is NOT being repeated...
  {
    if (alerts_this_bar < ALERTS_PER_BAR ||         // ... and if the number of alerts this bar has not
exceeded the limit
        ALERTS_PER_BAR == 0)                       // or if the limit is zero...
    {
      alertTime[pos] = current_time;                // ... then update the most recent alert time
      alertBar[pos] = current_bar;                 // update the most recent alert bar
      alertType[pos] = 0;                          // update the most recent alert type to CLOSE POSITION
      alertCount[pos]++;                           // increment the number of alerts created in the current bar


      if (DISPLAY_ALERTS)
        Alert(currencyPair + ":    " + trade_status
          + "\n\nTimeframe: "  + printableTimeframe()
          + "\nPrice:   "    + print_price
          + "\nSpread:   "   + DoubleToStr(spread,2)
          + "\nCause:   "    + AlertCausedBy);     // generate a CLOSE POSITION alert.
    }
  }


  else if (current_time - latest_alert_time > WAIT_AFTER_ALERT)
  {
    // CORRECTIONAL BUY ORDER FOR CLOSE POSITION
    // If a close position alert for a buy was created in the current bar, and close order conditions
    // are no longer valid within the bar, a correctional buy alert is created.
    if (open == 1 && latest_alert_type == 0 &&     // if the market is open and the CLOSE POSITION order
is most recent order
          latest_alert_bar == current_bar &&       // and if the most recent alert has been made within the
current bar
          trade_status == "WAIT"           &&      // and if there are no alert conditions met
          latest_open_alert >= 1)                  // and if the most recent opening order was BUY...
    {
      if (alerts_this_bar < ALERTS_PER_BAR ||         // ... and if the number of alerts this bar has not
exceeded the limit
        ALERTS_PER_BAR == 0)                       // or if the limit is zero...
      {
        alertTime[pos] = current_time;                  // ... then update the most recent alert time
        alertType[pos] = 2;                            // update the most recent alert type to CORRECTIONAL
BUY
        openAlert[pos] = 2;                            // update the most recent open order type to
CORRECTIONAL BUY
        alertCount[pos]++;                             // increment the number of alerts created in the current
bar
        trade_status = "BUY";                            // Correct trade_status to fit condition
        AlertCausedBy = "Correct for invalid Close Position."; // Declare condition that caused the alert.

        if (DISPLAY_ALERTS)
```

```
            Alert(currencyPair + ":    " + trade_status
              + "\n\nTimeframe: "  + printableTimeframe()
              + "\nPrice:   "    + print_price
              + "\nSpread:   "   + DoubleToStr(spread,2)
              + "\nCause:   "    + AlertCausedBy);       // generate a BUY alert.
          }
        }

        // CORRECTIONAL SELL ORDER FOR CLOSE POSITION
        // If a close position alert for a sell was created in the current bar, and close order conditions
        // are no longer valid within the bar, a correctional sell alert is created.
        else if (open == 1 && latest_alert_type == 0 &&     // if the market is open and the CLOSE POSITION
order is most recent order
            latest_alert_bar == current_bar &&         // and if the most recent alert has been made within the
current bar
            trade_status != "WAIT"           &&       // and if there are no alert conditions met
            latest_open_alert <= -1)                  // and if the most recent opening order was SELL...
        {
          if (alerts_this_bar < ALERTS_PER_BAR ||         // ... and if the number of alerts this bar has not
exceeded the limit ...
            ALERTS_PER_BAR == 0)                      // or if the limit is zero...
          {
            alertTime[pos] = current_time;              // ... then update the most recent alert time
            alertType[pos] = -2;                      // update the most recent alert type to CORRECTIONAL SELL
            openAlert[pos] = -2;                      // update the most recent open order type to
CORRECTIONAL BUY
            alertCount[pos]++;                        // increment the number of alerts created in the current bar
            trade_status = "SELL";                    // correct trade status to fit condition
            AlertCausedBy = "Correct for invalid Close Position";

            if (DISPLAY_ALERTS)
              Alert(currencyPair + ":    " + trade_status
                + "\n\nTimeframe: "  + printableTimeframe()
                + "\nPrice:   "    + print_price
                + "\nSpread:   "   + DoubleToStr(spread,2)
                + "\nCause:   "    + AlertCausedBy);   // generate a SELL alert.
          }
        }


        // CORRECTIONAL CLOSE POSITION ORDER FOR BUY OR SELL
        // If a buy or sell alert for a sell was created in the current bar, and close order conditions
        // are no longer valid within the bar, a correctional sell alert is created.
        else if (latest_alert_type == 1    ||       // If the BUY order is the most recent order OR
            latest_alert_type == -1)          // If the SELL order is most recent order
        {
          if (latest_alert_bar == current_bar &&       // and if the most recent alert has been made within the
current bar
            trade_status == "WAIT")                   // and if there are no alert conditions met
          {
            if (alerts_this_bar < ALERTS_PER_BAR ||         // ... and if the number of alerts this bar has not
exceeded the limit ...
            ALERTS_PER_BAR == 0)                      // or if the limit is zero...
            {
              alertTime[pos] = current_time;              // ... then update the most recent alert time
              alertType[pos] = 0;                     // update the most recent alert type to SELL
```

```
            alertCount[pos]++;                        // increment the number of alerts created in the current bar
            trade_status = "CLOSE POSITION";                        //
            AlertCausedBy = "Correct for invalid opening order.";

            if (DISPLAY_ALERTS)
              Alert(currencyPair + ":    " + trade_status
                + "\n\nTimeframe: "  + printableTimeframe()
                + "\nPrice:   "    + print_price
                + "\nSpread:   "   + DoubleToStr(spread,2)
                + "\nCause:   "    + AlertCausedBy);   // generate a CLOSE POSITION alert.
          }
        }
      }
    }
}



//+------------------------------------------------------------------+
//| Global MA Analysis alertCondition function                       |
//+------------------------------------------------------------------+
//RUNS CONDITIONS AND RETURNS THE TRADE COMMAND OF "BUY", "SELL", "CLOSE
POSITION", OR "WAIT"
string alertCondition(string  currencyPair)
{
  AlertCausedBy = "VOID";

  //determines pip decimal for the currency pair
  double PIPdecimal = 10000.0;              // default multiplier to display pips.
  if (StringFind(currencyPair,"JPY",0) >= 0) // if the currency pair compares to JPY...
    PIPdecimal = 100.0;                     // ...changes multiplier to represent pips for JPY

  double buy_cross_buffer_pips = BUY_CROSS_BUFFER / PIPdecimal; // converts the
CROSS_BUFFER int to the pips decimal
  double sell_cross_buffer_pips = SELL_CROSS_BUFFER / PIPdecimal; // converts the
CROSS_BUFFER int to the pips decimal
  double close_cross_buffer_pips = CLOSE_CROSS_BUFFER / PIPdecimal; // converts the
CROSS_BUFFER int to the pips decimal

  int shortEMA_period = EMA1;
  int longEMA_period  = EMA1;

  //determines which EMA is larger
  if (EMA1 < EMA2)
    longEMA_period  = EMA2;
  else
    shortEMA_period = EMA2;

  // calculates exponential moving average for the two EMA timeframes for current bar.
  double shortEMA_current = iMA(currencyPair, Period(), shortEMA_period, 0, MODE_EMA, 0,
HISTORY_SHIFT);
  double longEMA_current  = iMA(currencyPair, Period(), longEMA_period , 0, MODE_EMA, 0,
HISTORY_SHIFT);

  // calculates exponential moving average for the two EMA timeframes for previous bar
```

```
  double shortEMA_previous = iMA(currencyPair, Period(), shortEMA_period, 0, MODE_EMA, 0,
HISTORY_SHIFT+1);
  double longEMA_previous  = iMA(currencyPair, Period(), longEMA_period , 0, MODE_EMA, 0,
HISTORY_SHIFT+1);

  // calculates simple moving average for SMA and closing price of the bar for later comparison
  double SMA_current    = iMA(currencyPair, Period(), SMA, 0, MODE_SMA, 0, HISTORY_SHIFT);
  double SMA_previous   = iMA(currencyPair, Period(), SMA, 0, MODE_SMA, 0, HISTORY_SHIFT+1);

  double price_current  = iClose(currencyPair, Period(), HISTORY_SHIFT+1);
  double price_previous = iClose(currencyPair, Period(), HISTORY_SHIFT+1);

  // BUY CONDITION 1
  // - price crosses above SMA (including cross_buffer safe-guard)
  // - shortEMA is above longEMA (including cross_buffer safe-guard)
  if (price_current > SMA_current + buy_cross_buffer_pips &&
      price_previous < SMA_previous &&
        shortEMA_current > longEMA_current + buy_cross_buffer_pips)
        {
          AlertCausedBy = "Price crossed above SMA" + DoubleToStr(SMA,0) + ".";
          return ("BUY");
        }

  // BUY CONDITION 2
  // - shortEMA crosses above longEMA (including cross_buffer safe-guard)
  // - price is above SMA (including cross_buffer safe-guard)
  if (shortEMA_current > longEMA_current + buy_cross_buffer_pips &&
      shortEMA_previous < longEMA_previous &&
        price_current > SMA_current + buy_cross_buffer_pips)
        {
          AlertCausedBy = "EMA" + DoubleToStr(shortEMA_period,0) + " crossed above EMA" +
DoubleToStr(longEMA_period,0) + ".";
          return ("BUY");
        }

  // SELL CONDIITON 1
  // - price crosses under SMA (including cross_buffer safe-guard)
  // - shortEMA is below longEMA (including cross_buffer safe-guard)
  if (price_current < SMA_current - sell_cross_buffer_pips &&
      price_previous > SMA_previous &&
        shortEMA_current < longEMA_current - sell_cross_buffer_pips)
        {
          AlertCausedBy = "Price crossed below SMA" + DoubleToStr(SMA,0) + ".";
          return ("SELL");
        }

  // SELL CONDITION 2
  // - shortEMA crosses under longEMA (including cross_buffer safe-guard)
  // - price is below SMA (including cross_buffer safe-guard)
  if (shortEMA_current < longEMA_current - sell_cross_buffer_pips &&
      shortEMA_previous > longEMA_previous &&
        price_current < SMA_current - sell_cross_buffer_pips)
        {
          AlertCausedBy = "EMA" + DoubleToStr(shortEMA_period,0) + " crossed below EMA" +
DoubleToStr(longEMA_period,0) + ".";
          return ("SELL");
```

```
        }

  // CLOSE POSITION CONDITION 1
  // - shortEMA crosses above longEMA (including cross_buffer safe-guard)
  // - price is below SMA50 (including cross_buffer safe-guard)
  if (shortEMA_current > longEMA_current  + close_cross_buffer_pips&&
      shortEMA_previous < longEMA_previous &&
        price_current < SMA_current + close_cross_buffer_pips)
        {
          AlertCausedBy = "EMA" + DoubleToStr(shortEMA_period,0) + " crossed above EMA" +
DoubleToStr(longEMA_period,0) + ".";
          return ("CLOSE POSITION");
        }

  // CLOSE POSITION CONDITION 2
  // - shortEMA crosses under longEMA (including cross_buffer safe-guard)
  // - price is above SMA50 (including cross_buffer safe-guard)
  if (shortEMA_current < longEMA_current  - close_cross_buffer_pips&&
      shortEMA_previous > longEMA_previous &&
        price_current > SMA_current - close_cross_buffer_pips)
        {
          AlertCausedBy = "EMA" + DoubleToStr(shortEMA_period,0) + " crossed below EMA" +
DoubleToStr(longEMA_period,0) + ".";
          return ("CLOSE POSITION");
        }


  // CLOSE POSITION CONDITION 3
  // - price crosses under SMA (including cross_buffer safe-guard)
  // - shortEMA is above longEMA (including cross_buffer safe-guard)
  if (price_current < SMA_current - close_cross_buffer_pips &&
      price_previous > SMA_previous &&
        shortEMA_current > longEMA_current + close_cross_buffer_pips)
        {
          AlertCausedBy = "Price crossed below SMA" + DoubleToStr(SMA,0) + ".";
          return ("CLOSE POSITION");
        }

  // CLOSE POSITION CONDITION 4
  // - price crosses above SMA (including cross_buffer safe-guard)
  // - shortEMA is below longEMA (including cross_buffer safe-guard)
  if (price_current > SMA_current + buy_cross_buffer_pips &&
      price_previous < SMA_previous &&
        shortEMA_current > longEMA_current + buy_cross_buffer_pips)
        {
          AlertCausedBy = "Price crossed above SMA" + DoubleToStr(SMA,0) + ".";
          return ("CLOSE POSITION");
        }


  return ("WAIT");
}
/**************************************************************************************************
--------------------------------------------------------------------------------------------------
END                          GMAA RUNNING ALERT FUNCTIONS                          END
--------------------------------------------------------------------------------------------------
```

**********/

Dynamic Pip Overview Indicator Code

```
//+------------------------------------------------------------------+
//|                                    Dynamic PIP Overview.mq4 |
//|                                                  Craig Nesbitt |
//|                                                canesbitt@wpi.edu |
//+------------------------------------------------------------------+
#property copyright "Craig Nesbitt"
#property link      "canesbitt@wpi.edu"

#define pairs 28 // total number of currency pairs
#define columns 18 // number of columns in the graph

#property indicator_chart_window

/*****************************************************************************************************
------------------------------------------------------------------------------------------------------
START                        DYNAMIC PIP OVERVIEW INFORMATION                        START
------------------------------------------------------------------------------------------------------
*****************************************************************************************************
```

PURPOSE:

The Dynamic PIP Overview indicator displays the change in value (measured in PIPs) of the 28 major currency pairs in the form of a chart. Each row of the chart displays the name of the currency pair as well as the status of that currency over 13 time frames:

- 1 Minute, 5 Minutes, 10 Minutes, 15 Minutes, 30 Minutes, 1 hour, 2 hours, 4 hours, daily, weekly, monthly, quarterly, and annually.

The currency pairs are displayed in descending order in accordance to the time frame specified by the user.

SETUP:

The Dynamic PIP Overview indicator does not require specific currency window or time frame to function. To set up the indicator:

1) Save the Dynamic PIP Overview MQL4 Source File to
C:\Program Files (x86)\FXCM MetaTrader 4\experts\indicators
OR Your computer's equivalent custom indicator location

To find where the custom indicators are accessed by the terminal:
a) Right click on an existing cutom indicator in the MetaTrader Terminal and select Modify
b) Once in the MetaEditor, open the file menu and select Save As.
c) The indicator file location is at the top of the Save As window.

2) Open a new window in the MetaTrader Terminal. The currency pair and time frame does not matter.

3) Right click on the window and select "Properties(F8)"

4) In the Colors tab, change the color of all options to Black

5) In the Common tab, ensure the "Chart on foreground" is unchecked

6) Select "Okay" to close properties. The window should now be solid black.

7) Open the Dynamic PIP Overview from the Custom Indicator Menu in the Navigator
If you have just saved the Dynamic PIP Overview to the correct folder,
and it has not yet shown up in the Terminal Navigator, Compile the
Dynamic PIP overview and press the Terminal button in the MetaEditor.

```
**************************************************************************************************
--------------------------------------------------------------------------------------------------
END                        DYNAMIC PIP OVERVIEW INFORMATION                        END
--------------------------------------------------------------------------------------------------
**************************************************************************************************/
```

```
/**************************************************************************************************
--------------------------------------------------------------------------------------------------
START                      EXTERNAL INPUTS AND GLOBAL VARIABLES                      START
--------------------------------------------------------------------------------------------------
**************************************************************************************************/

//////////////// EXTERNAL INPUTS ////////////////

extern int ranked_by_column = 6; // Which column the currency pairs are being ranked in accordance to.

// Chart Dimensions and Style.
extern int scaleX = 75,       // horizontal interval at which the cells are created
          scaleY = 22,       // vertical interval
          offsetX = 10,      // horizontal indent of all cells
          offsetY = 20,      // vertical indent
          fontSize = 10,     // title font size
          numberSize = 11;   // data font size

extern int history_shift = 0;         // number of bars in the history that the indicator is calculating


//////////////// GLOBAL VARIABLES ////////////////

extern color    fontColor = Yellow;          // Default Color of font
extern string   titleFont = "Arial Bold";    // Default Title font
extern string   monospace = "Courier New";   // Default font style


extern color    positive_color = Green;
extern color    negative_color = Maroon;

string TITLES [columns] = {"RANK", "SYMBOL", "1 MIN", "5 MIN" , "10 MIN", "15 MIN", "30 MIN",
```

```
                    "1 HR", "2 HR", "4 HR", "DAY", "WEEK", "MONTH", "QTR", "YEAR",
                    "TODAY", "CU.WK", "CU.MN"};

/*****************************************************************************************************
-----------------------------------------------------------------------------------------------
END                     EXTERNAL INPUTS AND GLOBAL VARIABLES                          END
-----------------------------------------------------------------------------------------------
*****************************************************************************************************/




/*****************************************************************************************************
-----------------------------------------------------------------------------------------------
START                     CURRENCY PAIR INFORMATION ARRAYS                          START
-----------------------------------------------------------------------------------------------
*****************************************************************************************************/


// Currency pair ID's of the 8 major currencies
string pairnames[pairs] = {"EURUSD", // pos = 0
                    "GBPUSD", // pos = 1
                    "USDCHF", // pos = 2
                    "USDJPY", // pos = 3
                    "AUDUSD", // pos = 4
                    "EURCHF", // pos = 5
                    "EURGBP", // pos = 6
                    "NZDUSD", // pos = 7
                    "USDCAD", // pos = 8
                    "AUDCAD", // pos = 9
                    "AUDCHF", // pos = 10
                    "CADJPY", // pos = 11
                    "CHFJPY", // pos = 12
                    "EURAUD", // pos = 13
                    "EURCAD", // pos = 14
                    "EURNZD", // pos = 15
                    "GBPAUD", // pos = 16
                    "GBPCAD", // pos = 17
                    "GBPCHF", // pos = 18
                    "GBPNZD", // pos = 19
                    "CADCHF", // pos = 20
                    "NZDCHF", // pos = 21
                    "NZDCAD", // pos = 22
                    "NZDJPY", // pos = 23
                    "AUDNZD", // pos = 24
                    "AUDJPY", // pos = 25
                    "EURJPY", // pos = 26
                    "GBPJPY"};// pos = 27



// safeguard pricepips (pips of the current pair
```

```
double pricePIPS[pairs];

// arrays of the number of PIPs moved in each timeframe corresponding to pairnames[] positions
double pairPIPS_1min[pairs];
double pairPIPS_5min[pairs];
double pairPIPS_10min[pairs];
double pairPIPS_15min[pairs];
double pairPIPS_30min[pairs];
double pairPIPS_1hr[pairs];
double pairPIPS_2hr[pairs];
double pairPIPS_4hr[pairs];
double pairPIPS_day[pairs];
double pairPIPS_week[pairs];
double pairPIPS_month[pairs];
double pairPIPS_qtr[pairs];
double pairPIPS_year[pairs];
double pairPIPS_today[pairs];
double pairPIPS_thisWK[pairs];
double pairPIPS_thisMN[pairs];



/*
  ALPHABETICICAL RANKING

  "AUDCAD", // pos = 9  . rank 1
  "AUDCHF", // pos = 10 . rank 2
  "AUDJPY", // pos = 25 . rank 3
  "AUDNZD", // pos = 24 . rank 4
  "AUDUSD", // pos = 4  . rank 5
  "CADCHF", // pos = 20 . rank 6
  "CADJPY", // pos = 11 . rank 7
  "CHFJPY", // pos = 12 . rank 8
  "EURAUD", // pos = 13 . rank 9
  "EURCAD", // pos = 14 . rank 10
  "EURCHF", // pos = 5  . rank 11
  "EURGBP", // pos = 6  . rank 12
  "EURJPY", // pos = 26 . rank 13
  "EURNZD", // pos = 15 . rank 14
  "EURUSD", // pos = 0  . rank 15
  "GBPAUD", // pos = 16 . rank 16
  "GBPCAD", // pos = 17 . rank 17
  "GBPCHF", // pos = 18 . rank 18
  "GBPJPY", // pos = 27 . rank 19
  "GBPNZD", // pos = 19 . rank 20
  "GBPUSD", // pos = 1  . rank 21
  "NZDCAD", // pos = 22 . rank 22
  "NZDCHF", // pos = 21 . rank 23
  "NZDJPY", // pos = 23 . rank 24
  "NZDUSD", // pos = 7  . rank 25
  "USDCAD", // pos = 8  . rank 26
  "USDCHF", // pos = 2  . rank 27
  "USDJPY", // pos = 3  . rank 28

*/
```

```
// array of the ranking of each currency corresponding to pairnames[] positions
// defaults to alphabetical order
int pair_rank[pairs] = {15,21,27,28,5,11,12,25,26,1,2,7,8,9,10,14,16,17,18,20,6,23,22,24,4,3,13,19};


// array of the currency pair names ranked in descending order.
string ranknames[pairs] = {"EURUSD", "GBPUSD", "USDCHF", "USDJPY", "AUDUSD", "EURCHF",
"EURGBP",
                     "NZDUSD", "USDCAD", "AUDCAD", "AUDCHF", "CADJPY", "CHFJPY", "EURAUD",
                     "EURCAD", "EURNZD", "GBPAUD", "GBPCAD", "GBPCHF", "GBPNZD", "CADCHF",
                     "NZDCHF", "NZDCAD", "NZDJPY", "AUDNZD", "AUDJPY", "EURJPY", "GBPJPY"};

// array of the currency pair PIPs sorted in descending order.
double rankPIPS[pairs];


/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
END                          CURRENCY PAIR INFORMATION ARRAYS                          END
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/









/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                          INDICATOR MAIN FUNCTIONS                          START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/


//+------------------------------------------------------------------+
//| Custom indicator initialization function                         |
//+------------------------------------------------------------------+
int init()
  {
   string short_name = "Dynamic PIP Overview";
   IndicatorShortName(short_name);

   buildGraph();  // Constructs an empty chart in the window.
   updateGraph(); // Fills chart with current data


   return(0);
  }
```

```
//+------------------------------------------------------------------+
//| Custom indicator deinitialization function                       |
//+------------------------------------------------------------------+
int deinit()
  {
//----
     // Titles (accross top)
   for (int title = 0; title < columns; title++)
   {
     ObjectDelete(TITLES[title]);
   }



   //Rank List (down side)
   for (int rank = 1; rank <= pairs; rank++)
   {
     ObjectDelete("Rank " + rank);

     for (title = 1; title < columns; title++)
     {
       //Currency pair information (fills graph)
       ObjectDelete (TITLES[title] + rank);
     }
   }
//----
   return(0);
}

//+------------------------------------------------------------------+
//| Custom indicator iteration function                              |
//+------------------------------------------------------------------+
int start()
  {
   updateGraph(); // Fills chart with current data

   return(0);
  }
//+------------------------------------------------------------------+


/**************************************************************************************************
--------------------------------------------------------------------------------------------------
END                          INDICATOR MAIN FUNCTIONS                              END
--------------------------------------------------------------------------------------------------
**************************************************************************************************/




/**************************************************************************************************
```

```
---------------------------------------------------------------------------------------------
START                        INITIALIZATION HELPER FUNCTIONS                          START
---------------------------------------------------------------------------------------------
*********************************************************************************************/


//+------------------------------------------------------------------+
//| Dynamic PIP Overview buildGraph function                         |
//+------------------------------------------------------------------+
// buildGraph() creates the objects for all cells in the graph
//       - helper function to init()
void buildGraph()
{
  int x = 0;

  // Titles (accross top)
  for (int title = 0; title < columns; title++)
  {
    ObjectCreate (TITLES[title], OBJ_LABEL,0,0,0,0,0);                          // Creates Title labels
    ObjectSet    (TITLES[title], OBJPROP_YDISTANCE, offsetY);                   // Places Title Y

    if (title <= 1)
      ObjectSet    (TITLES[title], OBJPROP_XDISTANCE, x*scaleX+offsetX);       // Places non-
decimal Title X
    else
      ObjectSet    (TITLES[title], OBJPROP_XDISTANCE, x*scaleX+offsetX + 20);  // Places decimal
Title X

    ObjectSetText(TITLES[title], TITLES[title], fontSize,titleFont,fontColor);  // Sets title


    if (title == 14)
      x+=2;
    else x++;
  }




  //Rank List (down side)
  for (int rank = 1; rank <= pairs; rank++)
  {
    x = 1;

    ObjectCreate ("Rank " + rank, OBJ_LABEL,0,0,0,0,0);                  // Creates rank  label
    ObjectSet    ("Rank " + rank, OBJPROP_XDISTANCE, offsetX);           // Places rank X coordinate
    ObjectSet    ("Rank " + rank, OBJPROP_YDISTANCE,rank*scaleY + offsetY);  // Places rank Y
coordinate
    ObjectSetText("Rank " + rank, " " + rank ,fontSize,titleFont,fontColor);     // Sets rank Font

    for (title = 1; title < columns; title++)
    {
      //Currency pair information (fills graph)
      ObjectCreate (TITLES[title] + rank, OBJ_LABEL,0,0,0,0,0);                  // Creates graph contents
label
```

```
      ObjectSet    (TITLES[title] + rank, OBJPROP_XDISTANCE, x*scaleX + offsetX);  // Places graph
contents X
      ObjectSet    (TITLES[title] + rank, OBJPROP_YDISTANCE,rank*scaleY + offsetY);    // Places graph
contents Y
      ObjectSetText(TITLES[title] + rank, CharToStr(110),numberSize,monospace,fontColor); // Sets
graph contents Font

      if (title == 14)
        x+=2;
      else x++;
    }
  }
}

/**************************************************************************************************
--------------------------------------------------------------------------------------------
END                        INITIALIZATION HELPER FUNCTIONS                          END
--------------------------------------------------------------------------------------------
**************************************************************************************************/




/**************************************************************************************************
--------------------------------------------------------------------------------------------
START                        UPDATE PIP FUNCTIONS                           START
--------------------------------------------------------------------------------------------
**************************************************************************************************/



//+------------------------------------------------------------------+
//| Dynamic PIP Overview updateGraph function                        |
//+------------------------------------------------------------------+
/* updateGraph traverses currency arrays,
    updates all currency pair calculations
    updates ranking of currency pairs
*/
void updateGraph()
{
  for (int pos = 0; pos < pairs ; pos++)
  {
    updatePair(pos); // updates currency pair in array positions
  }

  updateRank();     // updates currency pair ranking
  printUpdates();   // prints updated information to graph
}
```

```
//+------------------------------------------------------------------+
//| Dynamic PIP Overview updatePair function                         |
//+------------------------------------------------------------------+
/* updatePair takes an array position and runs calculations on corresponding pair
     calculates change in pips for each column for the given row
*/
void updatePair(int pos)
{
  int bars_included;    // Bars being included in periodicity
  double period_open;   // Opening price of the timeframe
  double period_close;  // Closing price of the timeframe

  //determines pip decimal for the currency pair
  double PIPmultiplier = 10000.0;               // default multiplier to display pips.
  if (StringFind(pairnames[pos],"JPY",0) >= 0) // if the currency pair compares to JPY...
    PIPmultiplier = 100.0;                     // ...changes multiplier to represent pips for JPY

   // calculates current price in PIPs to compare to when displaying the currency
   pricePIPS[pos] = PIPmultiplier * iClose (pairnames[pos],PERIOD_M1, history_shift);


   //--------------------------------------------------------------------------------
   // update 1min array position using past 1 bars in M1 periodicity.
   bars_included = 1;
   period_open   = PIPmultiplier * iClose (pairnames[pos],PERIOD_M1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift);
   // updates array
   pairPIPS_1min[pos] = period_close - period_open;

   //--------------------------------------------------------------------------------

   // update 5min array position using past 5 bars in M1 periodicity.
   bars_included = 5;
   period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift);
   // updates array
   pairPIPS_5min[pos] = period_close - period_open;

   //--------------------------------------------------------------------------------

   // update 10min array position using past 10 bars in M1 periodicity.
   bars_included = 10;
   period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift);
   // updates array
   pairPIPS_10min[pos] = period_close - period_open;

   //--------------------------------------------------------------------------------

   // update 15min array position using past 15 bars in M1 periodicity.
   bars_included = 15;
   period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M1, history_shift);
```

```
// updates array
pairPIPS_15min[pos] = period_close - period_open;

//------------------------------------------------------------------------------

// update 30min array position using past 6 bars in M5 periodicity.
bars_included = 6;
period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M5, history_shift + bars_included);
period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M5, history_shift);
// updates array
pairPIPS_30min[pos] = period_close - period_open;

//------------------------------------------------------------------------------

// update 1hr array position using past 12 bars in M5 periodicity.
bars_included = 12;
period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M5, history_shift + bars_included);
period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M5, history_shift);
// updates array
pairPIPS_1hr[pos] = period_close - period_open;

//------------------------------------------------------------------------------

// update 2hr array position using past 8 bars in M15 periodicity.
bars_included = 8;
period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M15, history_shift + bars_included);
period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M15, history_shift);
// updates array
pairPIPS_2hr[pos] = period_close - period_open;

//------------------------------------------------------------------------------

// update 4hr array position using past 8 bars in M30 periodicity.
bars_included = 8;
period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_M30, history_shift + bars_included);
period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_M30, history_shift);
// updates array
pairPIPS_4hr[pos] = period_close - period_open;

//------------------------------------------------------------------------------

// update daily array position using past 6 bars in H4 periodicity.
bars_included = 6;
period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_H4, history_shift + bars_included);
period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_H4, history_shift);
// updates array
pairPIPS_day[pos] = period_close - period_open;

//------------------------------------------------------------------------------

// update weekly array position using past 7 bars in D1 periodicity.
bars_included = 7;
period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_D1, history_shift + bars_included);
period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_D1, history_shift);
// updates array
pairPIPS_week[pos] = period_close - period_open;
```

```
   //------------------------------------------------------------------------------

   // update month array position using past 4 bars in W1 periodicity.
   bars_included = 4;
   period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_W1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_W1, history_shift);
   // updates array
   pairPIPS_month[pos] = period_close - period_open;

   //------------------------------------------------------------------------------

   // update quarterly array position using past 13 bars in W1 periodicity.
   bars_included = 13;
   period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_W1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_W1, history_shift);
   // updates array
   pairPIPS_qtr[pos] = period_close - period_open;

   //------------------------------------------------------------------------------

   // update annual array position using past 12 bars in MN1 periodicity.
   bars_included = 12;
   period_open   = PIPmultiplier * iClose(pairnames[pos],PERIOD_MN1, history_shift + bars_included);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_MN1, history_shift);
   // updates array
   pairPIPS_year[pos] = period_close - period_open;

   //------------------------------------------------------------------------------

   // update today array position using current bar of day periodicity.
   period_open   = PIPmultiplier * iOpen(pairnames[pos],PERIOD_D1, history_shift);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_D1, history_shift);
   // updates array
   pairPIPS_today[pos] = period_close - period_open;

   //------------------------------------------------------------------------------

   // update current week array position using current bar of week periodicity.
   period_open   = PIPmultiplier * iOpen(pairnames[pos],PERIOD_W1, history_shift);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_W1, history_shift);
   // updates array
   pairPIPS_thisWK[pos] = period_close - period_open;

   //------------------------------------------------------------------------------

   // update current month array position using current bar of month periodicity.
   period_open   = PIPmultiplier * iOpen(pairnames[pos],PERIOD_MN1, history_shift);
   period_close  = PIPmultiplier * iClose(pairnames[pos],PERIOD_MN1, history_shift);
   // updates array
   pairPIPS_thisMN[pos] = period_close - period_open;

   //------------------------------------------------------------------------------

}
```

```
/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
END                          UPDATE PIP FUNCTIONS                                  END
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/




/*****************************************************************************************************
-----------------------------------------------------------------------------------------------------
START                        COLUMN RANKING FUNCTIONS                              START
-----------------------------------------------------------------------------------------------------
*****************************************************************************************************/


//+------------------------------------------------------------------+
//| Dynamic PIP Overview updateRank function                         |
//+------------------------------------------------------------------+
/* - applies updated values to rank arrays,
   - sorts the rank arrays,
   - updates rank position of pair_rank
*/
void updateRank()
{
   if (ranked_by_column != 0)
   {

      // Applies updated values to rank arrays.
      for (int rank_pos = 0; rank_pos < pairs; rank_pos++)
      {
         int match_found = 0; // nested for loop break for efficiency

         for (int pair_pos = 0; pair_pos < pairs && match_found == 0; pair_pos++)
         {
            if (pairnames[pair_pos] == ranknames[rank_pos])
            {
               rankPIPS[rank_pos] = rankedPIPvalue(pair_pos); // applies new value to ordered array.
               match_found = 1;                  // ends nested for loop.
            }
         }
      }



      //sorts the rank arrays.
      int inorder = 0; // initially assumes the arrays are not ordered.
      for (int round = pairs-1; round > 0 && inorder == 0; round--)
      {
         inorder = 1;        // assumes pairs are in ranked order for the round
         rank_pos = 0;       // resets array traversing counter
```

```
      while (rank_pos < round)
      {
        if (rankPIPS[rank_pos] >= rankPIPS[rank_pos+1])
          rank_pos++;          // rank arrays are in order

        else
        {
          inorder = 0;         // rank arrays are not in order.

          //swaps the positions
          string temp_name = ranknames[rank_pos];
          int temp_PIPS = rankPIPS [rank_pos];

          ranknames[rank_pos] = ranknames[rank_pos + 1];
          rankPIPS [rank_pos] = rankPIPS [rank_pos + 1];

          ranknames[rank_pos + 1] = temp_name;
          rankPIPS [rank_pos + 1] = temp_PIPS;
        }
      }
    }



    //matches new rank order to currency pairs
    for (pair_pos = 0; pair_pos < pairs; pair_pos++)
    {
      match_found = 0; // nested for loop break for efficiency

      for (rank_pos = 0; rank_pos < pairs && match_found == 0; rank_pos++)
      {
        if (pairnames[pair_pos] == ranknames[rank_pos])
        {
          pair_rank[pair_pos] = rank_pos + 1; // applies the rank number to the currency pair.
          match_found = 1;                   // ends nested for loop.
        }
      }
    }
  }
}



//+------------------------------------------------------------------+
//| Dynamic PIP Overview rankedPIPvalue helper function              |
//+------------------------------------------------------------------+
/* returns PIP value according to ranked column */
double rankedPIPvalue(int pos)
{
  switch (ranked_by_column)
  {
    // column 1: rank by 1 minute
    case 1:
      return (pairPIPS_1min[pos]);

    // column 2: rank by 5 minutes
```

```
case 2:
  return (pairPIPS_5min[pos]);

// column 3: rank by 10 minutes
case 3:
  return (pairPIPS_10min[pos]);

// column 4: rank by 15 minutes
case 4:
  return (pairPIPS_15min[pos]);

// column 5: rank by 30 minutes
case 5:
  return (pairPIPS_30min[pos]);

// column 6: default

// column 7: rank by 2 hours
case 7:
  return (pairPIPS_2hr[pos]);

// column 8: rank by 4 hours
case 8:
  return (pairPIPS_4hr[pos]);

// column 9: rank by day
case 9:
  return (pairPIPS_day[pos]);

// column 10: rank by week
case 10:
  return (pairPIPS_week[pos]);

// column 11: rank by month
case 11:
  return (pairPIPS_month[pos]);

// column 12: rank by quarter
case 12:
  return (pairPIPS_qtr[pos]);

// column 13: rank by year
case 13:
  return (pairPIPS_year[pos]);

// column 14: rank by today
case 14:
  return (pairPIPS_today[pos]);

// column 15: rank by current week
case 15:
  return (pairPIPS_thisWK[pos]);

// column 16: rank by current month
case 16:
  return (pairPIPS_thisMN[pos]);
```

```
      // default ranked column: 1 Hour
      default:
        return (pairPIPS_1hr[pos]);
    }
}


/**************************************************************************************************
-------------------------------------------------------------------------------------------
END                          COLUMN RANKING FUNCTIONS                          END
-------------------------------------------------------------------------------------------
**************************************************************************************************/




/**************************************************************************************************
-------------------------------------------------------------------------------------------
START                          PRINT FUNCTIONS                          START
-------------------------------------------------------------------------------------------
**************************************************************************************************/

//+------------------------------------------------------------------+
//| Dynamic PIP Overview printUpdates function                       |
//+------------------------------------------------------------------+
// Updates the objects on the graph in the ranked order.
void printUpdates()
{
   string loading_message = "N/A";
   color  loading_color   = DarkSlateGray;

  //Rank List (down side)
   for (int pos = 0; pos < pairs; pos++)
   {
      int title = 1;

       //---------------------------------------------------------------------------------------


      // currency pair name
      ObjectSetText(TITLES[title] + pair_rank[pos], pairnames[pos],fontSize,titleFont,White); // updates rank
position pair name

      title++;

      //---------------------------------------------------------------------------------------


      // currency pair 1Minute column
      if (pairPIPS_1min[pos] >= 0)
      {
```

```
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_1min[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_1min[pos]),numberSize,monospace,positive_color);
    }

    // updates rank position pair 1min value (negative)
    else ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_1min[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //------------------------------------------------------------------------------------------------


    // currency pair 5 Minute column
    if (pairPIPS_5min[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_5min[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_5min[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair 5min value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_5min[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //------------------------------------------------------------------------------------------------


    // currency pair 10 Minute column
    if (pairPIPS_10min[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_10min[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_10min[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair 10min value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_10min[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;
```

```
    //-------------------------------------------------------------------------------------------------


    // currency pair 15 Minute column
    if (pairPIPS_15min[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_15min[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_15min[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair 15min value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_15min[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //-------------------------------------------------------------------------------------------------


    // currency pair 30 Minute column
    if (pairPIPS_30min[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_30min[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_30min[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair 30min value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_30min[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //-------------------------------------------------------------------------------------------------


    // currency pair 1 hour column
    if (pairPIPS_1hr[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_1hr[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_1hr[pos]),numberSize,monospace,positive_color);
```

```
    }
    else
      // updates rank position pair 1hr value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_1hr[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //---------------------------------------------------------------------------------------


    // currency pair 2 hour column
    if (pairPIPS_2hr[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_2hr[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_2hr[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair 2hr value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_2hr[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //---------------------------------------------------------------------------------------


    // currency pair 4 hour column
    if (pairPIPS_4hr[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_4hr[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_4hr[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair 4hr value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_4hr[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //---------------------------------------------------------------------------------------


    // currency pair daily column
    if (pairPIPS_day[pos] >= 0)
    {
```

```
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_day[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_day[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair daily value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_day[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //---------------------------------------------------------------------------------------------


    // currency pair weekly column
    if (pairPIPS_week[pos] >= 0)
     {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_week[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_week[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair weekly value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_week[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //---------------------------------------------------------------------------------------------


    // currency pair monthly column
    if (pairPIPS_month[pos] >= 0)
     {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_month[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_month[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair monthly value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_month[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;
```

```
    //--------------------------------------------------------------------------------------------------


    // currency pair quarterly column
    if (pairPIPS_qtr[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_qtr[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_qtr[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair quarterly value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos], dec_align(pairPIPS_qtr[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //--------------------------------------------------------------------------------------------------


    // currency pair annual column
    if (pairPIPS_year[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_year[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_year[pos]),numberSize,monospace,positive_color);
    }
    else
      // updates rank position pair annual value (negative)
      ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_year[pos]*(-
1)),numberSize,monospace,negative_color);

    title++;

    //--------------------------------------------------------------------------------------------------


    // currency pair current day column
    if (pairPIPS_today[pos] >= 0)
    {
      //checks that display is valid
      if (pricePIPS[pos] == pairPIPS_today[pos])
        ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
      // updates rank position pair 1min value (positive)
      else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_today[pos]),numberSize,monospace,positive_color);
```

```
      }
      else
         // updates rank position pair current day value (negative)
         ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_today[pos]*(-
1)),numberSize,monospace,negative_color);

      title++;

      //------------------------------------------------------------------------------------------


      // currency pair current month column
      if (pairPIPS_thisWK[pos] >= 0)
      {
         //checks that display is valid
         if (pricePIPS[pos] == pairPIPS_thisWK[pos])
            ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
         // updates rank position pair 1min value (positive)
         else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_thisWK[pos]),numberSize,monospace,positive_color);
      }
      else
         // updates rank position pair current week value (negative)
         ObjectSetText(TITLES[title] + pair_rank[pos],dec_align(pairPIPS_thisWK[pos]*(-
1)),numberSize,monospace,negative_color);

      title++;

      //------------------------------------------------------------------------------------------


      // currency pair current month column
      if (pairPIPS_thisMN[pos] >= 0)
      {
         //checks that display is valid
         if (pricePIPS[pos] == pairPIPS_thisMN[pos])
            ObjectSetText(TITLES[title] + pair_rank[pos], loading_message
,numberSize,monospace,loading_color);
         // updates rank position pair 1min value (positive)
         else ObjectSetText(TITLES[title] + pair_rank[pos],
dec_align(pairPIPS_thisMN[pos]),numberSize,monospace,positive_color);
      }
      else
         // updates rank position pair current month value (negative)
         ObjectSetText(TITLES[title] + pair_rank[pos], " " + dec_align(pairPIPS_thisMN[pos]*(-
1)),numberSize,monospace,negative_color);

      title++;

      //------------------------------------------------------------------------------------------


   }
}
```

```
//+------------------------------------------------------------------+
//| Dynamic PIP Overview dec_align helper function                   |
//+------------------------------------------------------------------+
/* converts double to string,
 * adds spaces in front of string to align decimal
 * returns string to be printed.
 */
string dec_align(double pips)
{

   string pipstring = DoubleToStr(pips, 1);
   int count = 4 - StringFind(pipstring,".",0);

   string aligned;

   while (count > 0)
   {
      aligned = aligned + " ";
      count--;
   }

   aligned = aligned + pipstring;

   return (aligned);
}



/**************************************************************************************************
--------------------------------------------------------------------------------------------------
END                                PRINT FUNCTIONS                                 END
--------------------------------------------------------------------------------------------------
**************************************************************************************************/
```