Information in this document is subject to change without notice.

THE TRADING SYSTEMS IN THIS BOOK ARE EXAMPLES ONLY, AND HAVE BEEN INCLUDED SOLELY FOR EDUCATIONAL PURPOSES. OMEGA RESEARCH DOES NOT RECOMMEND THAT YOU USE ANY SUCH TRADING SYSTEM, AS THE USE OF ANY SUCH TRADING SYSTEM DOES NOT GUARANTEE THAT YOU WILL MAKE PROFITS, INCREASE PROFITS, OR MINIMIZE LOSSES. THE SOLE INTENDED USES OF THE TRADING SYSTEMS INCLUDED IN THIS BOOK ARE TO DEMONSTRATE THE WAYS IN WHICH EASYLANGUAGE CAN BE USED TO DESIGN PERSONAL TRADING SYSTEMS AND TO SHOW SOME EXAMPLES OF HOW CERTAIN POPULAR, WELL-KNOWN TRADING STRATEGIES MAY BE INCORPORATED INTO PERSONAL TRADING SYSTEMS. OMEGA RESEARCH, INC. IS NOT ENGAGED IN RENDERING ANY INVESTMENT OR OTHER PROFESSIONAL ADVICE. IF INVESTMENT OR OTHER PROFESSIONAL ADVICE IS REQUIRED, THE SERVICES OF A COMPETENT PROFESSIONAL SHOULD BE SOUGHT.

Contents

INTRODUCTION

Welcome to Volume 2

Welcome to Volume 2 of the *Omega Research System Trading and Development Club*, the newest learning tool offered by Omega Research to help you make the most of your trading potential.

This second volume of the *Omega Research System Trading and Development Club* contains 10 new systems we've created to help you get started developing your own systems. It also includes a new appendix, Appendix A, titled, "Volume in Review," which contains feedback on and corrections, when necessary, to the previous volume, which in this case is Volume 1. Those wanting additional discussion on the systems will benefit from this appendix.

The systems presented in this volume incorporate popular concepts such as the Head and Shoulders pattern, price and volume relationships, Bollinger Bands, and ADX. As in the first volume, we identify the most common problems with these types of systems and provide techniques to overcome them as well as present different ways of using them. By reviewing and testing these 10 systems, you'll be able to understand these techniques and use them, or a variation of them, in your own systems.

This book walks you step-by-step through the process we went through to develop these systems, from coming up with a feasible trading idea, to writing down your trading rules, to writing them in EasyLanguage, to taking into consideration money management and risk control factors. Our goal is for you to understand why we wrote these systems the way we did.

We recommend you study the EasyLanguage techniques we used, look at the System Report, read about the factors we took into account to deem that a system may have merit, and think about how you can combine some of these popular ideas with your own to develop a system *you* want to trade.

***IMPORTANT NOTICE*: The trading systems in this book are examples only, and have been included solely for educational purposes. Omega Research does not recommend that you use any such trading system, as the use of any such trading system does not guarantee that you will make profits, increase profits, or minimize losses. The sole intended uses of the trading systems included in this book are to demonstrate the ways in which EasyLanguage can be used to design personal trading systems and to show some examples of how certain popular, well-known trading strategies may be incorporated into personal trading systems.**

Contents at a Glance

This book begins by discussing some of the broader concepts of system development before moving on to the description of each system. We grouped the different types of systems together. The contents are:

IMPORTANT NOTE: *We suggest you read the book from front to back because there is instructional material in each section and it is not repeated throughout.*

Money-Back Guarantee

When you join the *Omega Research System Trading and Development Club*, approximately every two months you'll receive a new volume containing new systems along with the comprehensive handbook explaining in detail each system and the process used to develop it.

Keep in mind that the club is risk free—if you're not completely satisfied with any of the volumes you receive, simply return it within 30 days for an immediate refund. When you return a volume, unless you cancel your membership, you will still be a member of the club and receive the next volume. Of course, you can cancel your membership at any time without further obligation. But we're confident you'll find each volume worth your while!

If you have any questions regarding the club, please call Omega Research at (800) 439-7995, or (305) 551-9991 outside the US.

Additional Educational Services

Omega Research is committed to enhancing individual trading potential through quality education. To learn more about system trading, an Omega Research product, or EasyLanguage, visit our web site at **www.omegaresearch.com** or call **(800) 439-7995** (outside US 305-551-9991) and ask about the following educational services:

Video Courses

We offer a comprehensive video course, called *Becoming Fluent in EasyLanguage*. This course introduces you to EasyLanguage, starting with the very basics and taking you through writing actual

trading systems. It includes a follow-along workbook that contains exercises and real-life examples and applications.

Once you finish this course, you'll be able to write your own indicators, ShowMe and PaintBar studies, and trading systems. This course is perfect for those who want to learn EasyLanguage at their own pace. See for yourself, call now to place a risk-free order.

Workshops

Omega Research offers a variety of workshops on the products and technical analysis. Workshops are an excellent way to learn how to use the products, learn about technical analysis and system trading and/or EasyLanguage. Spend a day with a Product Training Specialist and exchange ideas with other users like yourself. All workshops provide a 100% satisfaction guarantee. Call now for more information or to register—space is limited!

EasyLanguage Resource Center

One of the best ways to learn is by example, and the EasyLanguage Resource Center on our web site is an excellent source of examples. In this Resource Center, we list all the analysis techniques—indicators and trading systems—published in the *Technical Analysis of Stocks and Commodities* magazine, as well as popular analysis techniques worth taking a look at. Access to this Resource Center is free of charge, feel free to download and review any of the analysis techniques and their descriptions. Our web site address is **www.omegaresearch.com**.

Getting Started

To begin reviewing your systems, transfer the analysis techniques into your TradeStation library and then apply the system you want to review to a chart. Use the System Report to view the system results and take a look at the EasyLanguage instructions by opening the system in the PowerEditor.

To transfer the analysis techniques into TradeStation:

1. Place the System Trading and Development Club CD in the CD-ROM drive.
2. Start the PowerEditor. In Windows 95, click **Start**, choose **Programs**, choose **Omega Research** and choose **TradeStation PowerEditor**. In Windows 3.x, choose **TradeStation PowerEditor** from the Omega Research program group.
3. In the PowerEditor, use the **File - Open** menu sequence.
4. Click **Transfer**.
5. Select the **Transfer analysis techniques FROM EasyLanguage Archive File** option and click **OK**.
6. Click **Scan**.
7. In the **Enter drive letter to scan** edit box, enter the drive letter for your CD-ROM drive (normally D), and click **OK**. The ELA file on the CD is placed in the list.
8. Choose **MARAPR98.ELA** from the list and click **OK**.
9. In the **Transfer** dialog box, select **Transfer All** and click **OK**.
10. Once the files are transferred and verified, a dialog box appears informing you that the transfer was performed successfully. Click **OK**.

For your convenience, the names of the systems in this volume all begin with STAD2. You can now open the systems in the PowerEditor and view the EasyLanguage instructions and/or apply them to a chart in TradeStation. You can remove your CD from the CD-ROM drive and store it in a safe place. As you apply the systems and work with them, refer to this book for detailed explanations of the systems and the EasyLanguage used to create them. For instructions on applying systems and viewing the System Report, please refer to your *TradeStation User's Manual*.

Note to SuperCharts Users: *To transfer the systems into SuperCharts, use the* ***Tools - QuickEditor*** *menu sequence and select* ***Transfer****. Keep in mind, however, that although you can apply the systems in SuperCharts, you will not be able to view the EasyLanguage instructions in the QuickEditor. This is because the systems were designed in the PowerEditor. Also, if you are using SuperCharts End of Day, some of the systems will not apply as they are designed for intraday trading. Since the purpose of the Club is to provide you with a learning tool, and viewing the EasyLanguage instructions is an essential part of this learning process, the use of this club for SuperCharts users is limited.*

Note to TradeStation or SuperCharts 3.x Users: *The systems for the Club were designed using TradeStation 4. As such, some of the features used, such as automatic drawing of trendlines and/or text, are not available in previous versions of TradeStation (or SuperCharts). An effort is made to provide a variety of systems that incorporate both long standing and new features, however, keep in mind that as new features are developed, we will naturally want to showcase and educate users on these features; therefore, users of the most recent version of our software will be able to make the most use of the Club.*

Obtaining Technical Support

Depending on your question, there are two resources at your disposal: the EasyLanguage Support Department and the STAD Club E-Mail Address.

EasyLanguage Support Department

The EasyLanguage Support Department provides EasyLanguage support via fax and is designed to help you troubleshoot an analysis technique or trading system you are currently working on. For example, if you are incorporating a trading system from the Club into your own and have a question about the implementation, the EasyLanguage Support Department can answer it.

Please keep in mind; however, that while this department can answer any EasyLanguage question, it cannot answer questions about the STAD Club specifically, such as the theory behind a system in the Club, why a system was developed a certain way, or why the system is not performing as you expect it to, etc.

Fax Number

(305) 221-6831

Be sure to include the following information in your fax:

- Name
- Security Block or Customer ID Number
- Telephone Number
- Fax Number
- Product you own
- EasyLanguage instructions you are working on
- Detailed description of your problem

Please allow 48 hours for a response.

STAD Club E-Mail Address

Another resource at your disposal is the STAD Club e-mail address:

stadclub@omegaresearch.com

Please send any comment, suggestion, or question regarding the systems in the Club to the STAD Club e-mail address, and each subsequent volume we will publish the most common suggestions and questions. ***IMPORTANT:*** *When you send a message to this e-mail address, you will not receive a response directly; your message will be reviewed and the answer incorporated into the next volume of the STAD Club, when applicable.*

CHAPTER 1

General System Development Concepts

As you are developing your systems, there are certain rules of thumb that you should follow. These are guidelines that our users have discovered through years of trial and effort, and we are providing them for you here.

Both Chapter 1 from Volume 1 and this chapter were adapted from Charlie Wright's book on system testing and development, *Trading as a Business*, currently available from Omega Research. It is the definitive book on system testing and development and puts years of successful experience at your fingertips.

This chapter discusses the idea of setup and entry, and explains how essential it is to use both together when developing your system. Most beginning system traders work on setups or entries only, completely ignoring an entire half of the system development puzzle. They discard valid ideas because they attempt to use them by themselves rather than as part of the setup and entry team, and deem them unsuccessful when in fact they could have been a valuable component of a successful system.

We strongly recommend that before you begin developing your systems, you read this chapter—it could save you hours of work and headaches!

In This Chapter

Developing Your Entry & Exit Rules

Once you have clearly defined the direction in which you are heading with your system, that is, once you've decided on the system type and market type you are going to trade, and have a feel for the types of patterns on which you want to capitalize, now is the time for a brainstorming session. You should sit down in front of your computer with TradeStation and start to develop the set of rules that actually make up your trading system.

Many traders at one time or another have become frustrated with system development. Not because they don't like it, but because they have run out of new ideas to test, or haven't found anything that works for them. For example, most traders have tested the Dual Moving Average Crossover System sometime in their trading career. Usually, the trader will look at this system and believe that the only thing to test is the length of the two averages; they will experiment with many different lengths for the averages. When they don't find any that work to their satisfaction, they discard the dual moving average system concept entirely, and move on to something else. They keep looking for that Holy Grail indicator that they can instantly make into a system. We have all been there, and have all discarded a lot of great ideas. However, more often than not, the discarding of an idea is a mistake.

For the most part, any indicator can be made into a profitable system. Yes, any indicator. When we discard the moving averages, it is usually a mistake because the moving averages by themselves only represent one half of the system development puzzle. This half is what we refer to as the "Setup" of a system. The second half of a system, the half that most traders ignore completely, is what we call the "Entry."

In this chapter, we'll talk about exactly what these two terms mean, and more importantly, how using them together can turn something as mundane as a moving average crossover into a promising new trading technique.

The Magic of SetUp and Entry

The secret to successful system development is to look at a method, or indicator, in an unconventional manner. The trick is to use it in a different and unique way. With Setup and Entry, you will look at system development in a completely different way. As you'll soon see, it can provide you with a whole new world of exciting possibilities and ideas to test. It will lift you out of the rut of simply optimizing standard indicators and give you a method of organizing your creativity.

THE SETUP

The Setup is the condition or set of conditions that are necessary prior to considering taking a position in the market. It is the indicator or group of indicators that tell you to get ready to buy or sell. Setups don't get you in the market, they simply make you aware that a trade is in the making.

Examples of setups for a trend-following system:

- A fast moving average crossing a slow moving average
- The ADX indicator in an up-trend
- Prices moving outside of a price channel

Examples of setups for a support and resistance system:

- The RSI moving into oversold territory (below 20) or into overbought territory (above 80)
- SlowD crossing SlowK when using the Stochastic Indicator
- Prices reaching the upper or lower line of a moving average envelope

Examples of setups for a volatility expansion system:

- An opening price gap over the high of the previous bar
- The current bar's range is greater than the average range of the last three bars
- The difference between two moving averages on the current bar is greater than the average difference of the last 10 bars

There are countless other indicators and conditions that could be used as setups. In the final analysis, you are limited only by your creativity. There is only one constraint that you should impose upon yourself. It is essential to recognize the type of system you are trying to develop and use the different indicators accordingly. You do not want to use a moving average crossover for a support and resistance system unless you are using it in a unique way. You would not choose to use the stochastic indicator for a trend-following system unless you had completely re-configured how it is used. Most system traders do not recognize that these indicators only set up the trade. They are unaware that there are a multitude of ways to actually get in the market once the setup has occurred. They are not aware that setups are only part of the equation and are not particularly profitable in and of themselves.

Beginning system developers get discouraged when they try to develop profitable systems from setups only. They quickly run out of ideas to test, because they use up all their ideas as setups without trying to combine them with various complementary entries. By trading only setups, you lose the added precision, accuracy and increased profitability of a system that uses both setup and entry. If trading setups by themselves worked, and was profitable, trading would be easy and all traders would be rich.

THE ENTRY

An entry is the signal by which the system purchases the contract in the market. It is the technique that you use to take a market position once the rules for the setup have been met. Entry selection is dependent on the type of setup you've designed. You may choose to trade a trend-following system, an S/R system, or a volatility expansion system. The entries are designed differently depending on the type of system you choose to trade. Many beginning traders devise systems that only trade entries. These are not as effective and are usually less profitable than systems that utilize both a setup and an entry. Systems based only on entries tend to have too many trades and a low percentage of profitable trades. There are two rules to which all entries must adhere:

Entry Rules

1. Prices should confirm the direction indicated by the Setup before a taking a position.

2. The Entry should guarantee that a system will capture every price move for which it is designed.

Entry Rule #1 requires prices to move in the expected direction before entering the market. If our setup indicates a long position, we would require the price action to move up in some specified manner before we would be comfortable taking a position. We want the price action to confirm the setup and force us into taking a position. For instance, let's assume that on today's close our setup has given us a long signal. We might require a breakout above the high of today's bar to confirm that the market is in bullish mode. With this breakout as a condition of entry, we have now required specific market action in the direction of the setup before we risk taking a market position.

Some examples of buy entries are:

- A buy stop on tick above the current bar's high
- A buy stop over the highest high of the last three bars

- Buy at market after several consecutive up closes
- Buy at market after a close over the previous bar's high
- Buy on a close that is greater than the open
- Buy on a stop, one tick above the last swing high
- Buy at the market on the close of a key reversal bar

When deciding on what type of signal to use as your entry, it is important to keep in mind the type of system you are trying to create. There are certain types of entries you don't want to use with setups because they have basic flaws that may allow the system to miss the big move.

Entry Rule #2 is to make sure that our entry guarantees that we will be in on every move that the system was designed to catch. The strength of this guarantee is the criteria by which you should judge the viability of all entries. An entry is flawed if there is even a slight chance that there could be a big move that the entry would miss. This is a very important system development principle that you should think about.

For example, you do not want to use a key reversal signal as the only entry for a trend-following system. There is absolutely no assurance that once the trend setup has occurred that a key reversal will follow. It is possible that after the moving averages have crossed, giving us a buy setup, the market may very well embark on a long up-trend without as much as once having a key reversal bar. Without the key reversal bar, we would not enter the market even though the trend setup has given us a signal. Without the key reversal bar, we will miss the big move. And missing the big move is the worst outcome for the trend trader.

Another example of a faulty entry is an entry that consists of three consecutive up or down closes. There is no guarantee that given a setup, this pattern will occur. The market may embark on a long trend without having three consecutive up or down closes in a row. It is possible that a trend-following system with this entry could miss the big move, and this possibility is a flaw in the system design that should be avoided. That is not to say however that key reversals or consecutive closes should not be used. You could compensate for their shortcomings as entries by including an additional entry or entries in the system that would serve as a backup. The entry or combination of entries must guarantee that the system will be in the market should any large trend develop.

Understanding Different Order Types

The only limit to creating viable entries is your creativity. There are potentially many techniques that make interesting entries. However, entries are also dependent on the type of order used. There are four basic orders that are commonly used for entries: Market orders, Stop Close Only orders, Stop orders, and Limit orders. Not all of these orders are available on every exchange. You should check the exchange you will be trading on for a list of the available order types.

Market Orders

A market order is used to enter the market without any restrictions on what the price should be. This order is commonly placed on the open of the day (market on open) or close of the day (market on close). However, market orders may also be placed anytime during the day by calling your broker and either buying or selling "at market."

Although market orders fulfill the criteria for Entry Rule #2, they are deficient because they violate Entry Rule #1. Market orders (market on close, market on open) are not entries at all. They are simply the obvious and easiest way to put on a trade. Market orders may be turned into viable entries by adding another condition to them that will signal an implied direction.

For example, an effective use of a market order would be to "buy tomorrow at market if the open tomorrow is greater than the high of today." This forces the market to indicate a direction,

presumably in the direction of the setup (in this case up) before we enter the market. A market order may be used to enter the market, but should always be used with at least one more condition in order to fulfill Entry Rule #1.

Stop Close Only Orders

Stop Close Only (SCO) orders are market orders with an important twist. The twist is that to enter long, the market must close above a price that we have pre-selected. For a sell, the market must close below our pre-selected price. An example is to buy a contract on the close at 856.30 SCO. This means that if the price closes at or above 856.30, the market will fill your order at the market. The idea is that with an SCO order, you have placed an important restriction on the market order, making it a viable entry. This forces the system developer to find a price that the market must close above (or below) before the system takes a position. By placing this restriction on a market order, we have turned it into a valid type of entry.

Stop Orders

The easiest way to create a valid entry is to use a stop order. By its nature, a stop requires the market to pass through a certain pre-selected price before a contract is bought or sold. Using a non-removable stop order is the best way to create innovative entries and confirm the entry rules. The reason stop orders are generally superior to SCO orders is that they guarantee that your system will enter the market regardless of when during the day the price is hit. You will not have to wait for the close and you may catch a big intraday move that would be lost if you waited for an SCO.

An example of using a stop order as an entry is the bar breakout entry. If today our setup turns bearish, we would place a sell order one tick below today's low. Unless prices move below this price, forcing a confirmation of the set-up, the system would not take a short position in the market. The same mechanics would hold true for a long signal. Stop orders that are not removed are also the best guarantee that the system will be in for the big move.

Placing a permanent sell stop (good until cancelled) below the current price provides the best assurance that you will be in on any move beyond that price. The floor brokers must fill your order as soon as they can once that price is hit. This guarantees you will be in on the move, although there is no guarantee as to the exact price (this difference between the stop and the fill price is slippage).

Limit Orders

Limit orders are the opposite of stop orders. By their nature, limit orders require prices to be traveling in a direction opposite the setup. The primary intent of the limit order is to place a resting buy order somewhere below the present market price. This is an attempt to pick off a lower and better price than where the market is currently. You may also place a resting sell order above the current price to sell at better than current prices.

Limit orders are primarily used in support and resistance systems and are generally not effective. Assume that the market is now trading at 258.00. The mechanics of a limit order are to place an order to buy a contract or share at 256.50 limit (or better) or sell at 258.50 or better (limit). This means that the floor brokers who are filling your order will only attempt to sell your contract at a price equal to or above 258.50. If the broker cannot sell the contract at or above the price, you will not be in the market. The same strategy is used with the limit buy order.

The limit order does not conform to Entry Rule #1 because it does not force prices to move in the direction of the set-up before entry. There could be a case in which a lower limit price was not reached before the market took off in a big up move. Even if the market by chance should hit this price, there still is no guarantee that the broker will be able to fill the price. Unlike a stop, which becomes a market order at the prescribed price, a limit order must be filled at or better than the prescribed price. The market may trade at that price for only one or two trades, and then move away quickly. If your broker is not the fastest broker, you may not get filled even though the market traded at your prescribed price.

Limit orders violate both Entry Rule #1 and Entry Rule #2, therefore, we do not recommend them for use as an entry.

Evaluating Each Component

The basic premise is that most new system developers do not organize their systems in this manner. They use either setups or entries, but not both. Using a setup or entry on its own generally does not work. The power comes when you combine the two.

A very effective method of evaluation is to compare the performance of each of the components of a system by themselves as well as the final system—take a look at how the setups perform on their own and at the profitability of the entry or entries by themselves. Compare these with the performance of the final system, which includes both the setups and entries. With this type of comparison, you are able to gauge what the different characteristics of each component are and what they add to the mix. This will let you determine whether or not the whole is greater than the sum of the parts.

What you should find is that the combined setup and entry system is the most profitable. You should find that setup and entry, and its rules, worked its magic and gave you a better system than any of the components could deliver by themselves.

By a better system, we mean one that you could trade with confidence. Ultimately, the question you have to ask yourself is could you trade this system? Could you trade and stick with the system you have designed? Just because the system is profitable and meets our system development criteria does not mean it is one we could or would want to trade. Just because it is profitable does not mean that you are emotionally able to trade it. So many traders create or purchase very profitable systems, but because their personality doesn't match the system, they still lose money, all the while lamenting the fact that they can't stick to the system.

Summary

Trading the setup and entry concept and making sure that you follow the rules gives far superior results when compared to trading either setups or entries by themselves. Using both a setup and an entry together enhances the performance and profitability of a system. Here's how you can summarize how you should think about setups and entries:

AIM WITH THE SETUP
PULL THE TRIGGER WITH THE ENTRY

Always use the concept of setup and entry to develop systems. There are two distinct parts to system writing and keeping these two components in mind will help you to organize your thoughts and design a sound strategy. Above all, this blueprint for system development opens up a whole new range of possibilities for you to test.

CHAPTER 2

Trending Systems

Trending systems are systems designed for trending markets—they have the following characteristics:

- The systems are designed never to miss the big move; they will either always be in the market or contain stop orders that will stop you into the market.
- They attempt to limit losses during the market's sideways mode; no system will make money in every market condition, but a good system will limit losses in market conditions for which it was not designed.
- Profits are concentrated in a few big trades; they have a low percentage of profitable trades. This makes them psychologically difficult to trade and underscores the importance of never missing a big move.

Even though trending systems are difficult to trade, they are popular—it's human nature to want to cash in on the big moves. In this chapter, we present three trending systems that differ in their approach but are all designed to capture big moves and limit losses during directionless and/or volatile phases.

In This Chapter

ADX Differential

The ADX is a unique indicator that shows the strength of the current trend; that is, whether or not the market is trending, and if so, how strong the trend is. The ADX Indicator does not indicate whether the trend is up or down, just whether or not it is trending and the strength of that trend. In fact, we consider the value itself of the ADX to be unimportant—what matters is the direction in which it is moving.

We have seen that markets that usually trend are likely to fall into non-trending periods characterized by a low or decreasing ADX. These non-trending periods are called sideways markets, congestion, or consolidation periods, and are usually a setup phase before the next trending period starts. When the value of the indicator is increasing, it means that the current trend is strong. Therefore, we decided to build a system around an ADX Indicator, using it to help us determine when the market is falling into a non-trending period and when it looks like the market is beginning a rally, either bullish or bearish.

To determine if the ADX is increasing or decreasing compared to the previous bar, we subtract the previous bar's ADX from the current bar's ADX. This enables us to quickly see where the ADX is going. Again, when the value of the indicator is increasing, it means that the current trend is strong, and when it is decreasing, or low, we assume the market is in a sideways mode, and we start looking for the expected bullish or bearish rally, characterized, again, by an increasing ADX.

Following the differential value between the current ADX and the previous ADX will sometimes give you a large number of consecutive crosses over and under zero, so instead of using the raw differential, we will smooth it out by producing a 4-bar exponential average of the differential, and use this value instead. For the purposes of this discussion, we'll refer to this exponential average of the differential as Delta Average.

Figure 1 shows an ADX indicator that we wrote applied to a chart. We wrote this ADX Indicator using the exact formula we use in our system. We wrote it as an indicator as well so that we could visually follow the system. The indicator is called STAD2: ADX Diff.

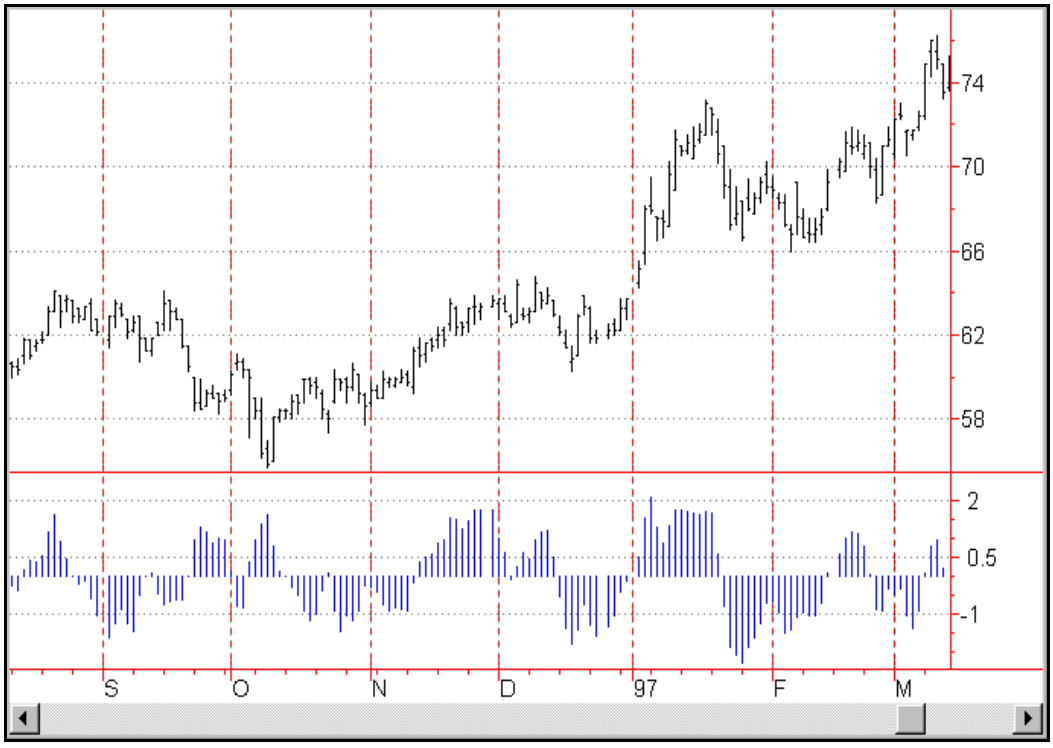


Figure 1. The STAD2: ADX Diff Indicator applied to a chart. Notice the rise and fall of the Delta Average compared to the price data.

When the Delta Average crosses under zero, we know that the market is entering a consolidation period. Conversely, when the Delta Average crosses over zero, we have a strong indication that the market is starting to rally.

When the Delta Average crosses either above or below zero, we will identify the channel the market is forming and thereby define the upper and lower band that we will use to buy and sell, respectively. We will buy when the upper channel is broken and sell when the lower channel is broken.

We build the upper band by adding half of a 4-bar average of the range to the close of the bar on which the Delta Average crosses above zero. Likewise, we build the lower band by subtracting half of a 4-bar average of the range to the close of the bar on which the Delta Average crosses under zero. Figure 2 shows both the ADX Diff Indicator and the ADX Differential system applied to the chart.

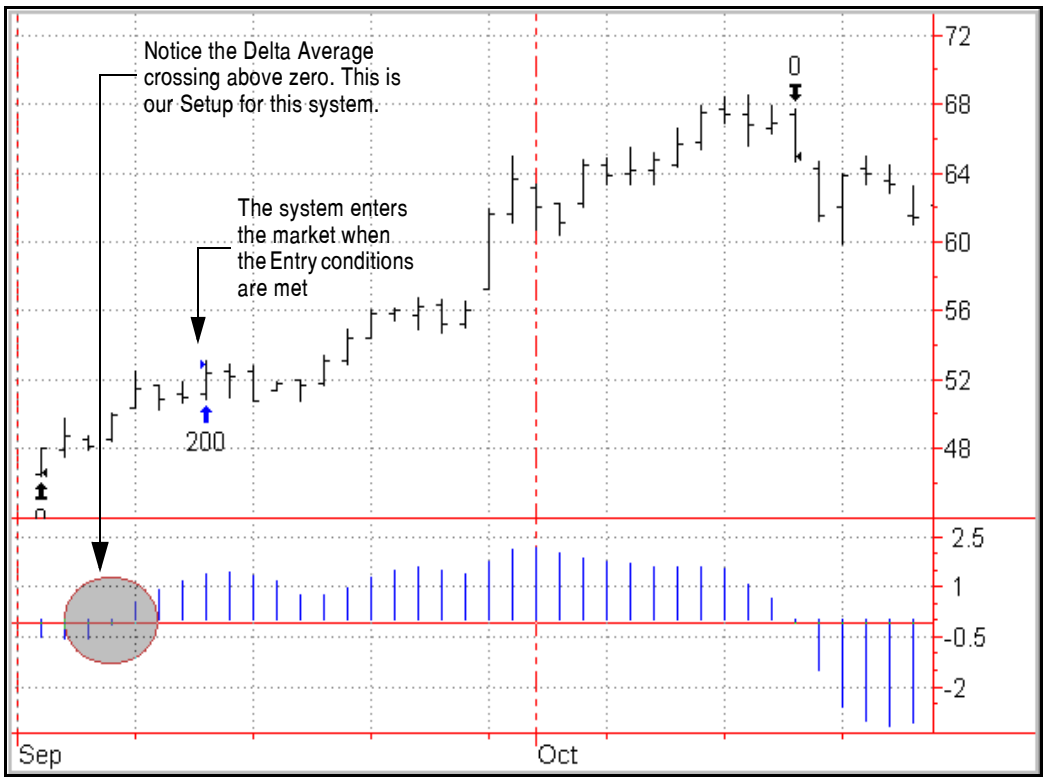


Figure 2. The ADX Diff Indicator and ADX Differential System applied to a chart

Once we determined how we were going to enter the market, we turned our attention to the method of exiting the market. We will work with a variation of a trailing stop exit technique we used in the a number of the systems in Volume 1 of STAD Club.

For our long positions, we will define an arbitrary point below the Low of our entry bar. In this system, we will subtract a 4-bar average of the range from the Low of the bar of entry, and use that as our exit point for the first bar. We will use inputs so that we can tighten this stop. Then, we will add a third of the distance between the Low of the bar and the previous exit point to the current exit point to determine the next bar's exit point. We will use this for all bars except our bar of entry. To cover our entry bar, we will risk the lowest low of the last 6 bars.

We will do the opposite for short positions. We will add a 4-bar average of the range to the High of the bar of entry and use this as a stop for the first bar. Thereafter, we will subtract a third of the distance from the previous stop and the High of the bar from the previous stop point to determine next bar's exit price. We will use this for all bars except our bar of entry. To cover our entry bar, we will risk the highest high of the last 6 bars.

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. The long and short entries reverse your position, whereas the exits close out your existing position and exit you from the market. We also performed some setup work, which involved calculating the ADX and the 4-bar exponential average of the ADX differential. The setup, entry and exits are described next.

Setup

a) Calculate the ADX value.

b) Calculate the ADX differential.

c) Calculate the 4-bar exponential average of the ADX differential.

Long Entries

a) If the exponential average of the ADX differential is under zero or crosses over zero, find a 4-bar average of the range, divide this average by 2, and then add the resulting value to the Close of the current bar.

b) Use the resulting value to place a stop order. This stop order will remain active for the next 10 bars.

Short Entries

a) Again, if the exponential average of the ADX differential is under zero or crosses over zero, find a 4-bar average of the range, divide this average by 2, and then subtract the resulting value from the Close.

b) Use the resulting value to place a stop order. This stop order will remain active for the next 10 bars.

Exits

a) At the bar of entry for a long position, we place an exit order for long positions at the lowest Low of the last 8 bars.

b) When in a long position, once the bar of entry is closed, we calculate the 4-bar average of the range, divide the average by 4, and then subtract this value from the Low of the current bar. This is our long exit price for bar 2.

c) When in a long position, for all bars after bar 2, we obtain the previously calculated exit price and add to it a third of the difference between the Low and the previous stop price. We repeat this operation at the end of every bar to give us the exit price for all subsequent bars.

d) At the bar of entry for a short position, we place an exit order for short positions at the highest High of the last 8 bars.

e) When in a short position, once the bar of entry is closed, we calculate the 4-bar average of the range and divide it by 4. Then, we add this value to the High of the current bar. This is our short exit price for bar 2.

f) When in a short position, for all bars after bar 2, we obtain the previously calculated exit price and subtract from it a third of the difference between the previous stop price and the High of the current bar. Repeating this operation at the end of every bar will give you the exit price for bar three and beyond.

g) We also place a money management protective stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts we are trading.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: ADX Differential System (STAD2: ADX Diff)

```
Input: ADXLen(14), AvgLen(4), CH(8);
Vars: UpCh(0), DnCh(0), MP(0), StopPrice(0), DeltaADX(0), DeltaAvg(0), LastOrderBar(0);

MP = MarketPosition;
Value1 = ADX(ADXLen);
DeltaADX = Value1 - Value1[1];
DeltaAvg = WAverage(DeltaADX , AvgLen );

{Checks for Entry conditions and calculates Entry Prices}
If (DeltaAvg < 0 or DeltaAvg crosses over 0 ) and BarNumber > 1 then Begin
        UpCh = Close + Average(Range, 4)/2;
        DnCh =  Close - Average(Range, 4)/2;
        LastOrderBar = BarNumber;
End;
If MP <> 0 then LastOrderBar = -999;

{Places BUY and SELL Orders as well as initial ExitShort and ExitLong Orders}
If BarNumber < LastOrderBar + CH then Begin
        Buy next bar at UpCh stop;
        Sell next bar at DnCh stop;
        If MP <> 1 then ExitLong ("LEntryStop") next bar at Lowest(Low,6) stop;
        If MP <> -1 then ExitShort ("SEntryStop") next bar at Highest(High,6) stop;
End;

{Calculates Stop Order price for Long Position}
If MP = 1 and MP[1] <> 1 then Begin
        LastOrderBar = -999;
        StopPrice = Low - Average(Range,4);
End;

{Calculates Stop Order price for Short Position}
If MP = -1 and MP[1] <> -1 then Begin
        StopPrice = High + Average(Range,4);
        LastOrderBar = -999;
End;

{Places Long Exit Order and calculates new Stop Order price}
If MP = 1 then Begin
        Exitlong ("LongTrailStop") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low-StopPrice)/3;
End;

{Places Short Exit Order and calculates new Stop Order price}
If MP = -1 then Begin
        Exitshort ("ShortTrailStop") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice-High)/3;
End;
```

Inputs

Inputs	Default	Description
ADXLen	14	Number of bars used to calculate ADX.
AvgLen	4	Number of bars used to calculate the average of the ADX differential.
CH	8	Number of bars to keep entry orders active after the adx diff. Average crosses over zero.

For a description of functions, variables and other EasyLanguage terminology, please refer to the *EasyLanguage User's Manual*.

In addition to these inputs, we define the following variables:

```
Vars: UpCh(0), DnCh(0), MP(0), StopPrice(0), DeltaADX(0), DeltaAvg(0), LastOrderBar(0);
```

First, we use the MarketPosition function to obtain the current market position and store this value in the variable MP. We also use the ADX function to calculate the value of the ADX and store this value in the variable Value1. Then, we calculate the ADX differential and store this value in the variable DeltaADX:

The function MarketPosition returns -1 when in a short position, 1 when in a long position, and 0 when flat, or no position.

```
MP = MarketPosition;
Value1 = ADX( ADXLen );
DeltaADX = Value1 - Value1[1];
```

Finally, we use the WAverage function to calculate the exponential average of the ADX differential, and store this value in the variable DeltaAvg:

```
DeltaAvg = WAverage( DeltaADX , AvgLen );
```

Entries

When the DeltaAvg is negative or on the bar at which it crosses over zero, we add and subtract half of the average of the range of the last 4 bars to the current Close price. These values become our long and short entry prices and we store them in the variables UpCh and DnCh for use in our BUY and SELL statements. We also store the bar number of the current bar in the variable LastOrderBar. We can then use the LastOrderBar variable to keep the orders active for any number of bars, as we will explain later.

```
If (DeltaAvg < 0 or DeltaAvg crosses over 0 ) and BarNumber > 1 then Begin
        UpCh = Close + Average(Range, 4)/2;
        DnCh =  Close - Average(Range, 4)/2;
        LastOrderBar = BarNumber;
End;
```

EasyLanguage Note: *BarNumber is a function that gives you the number of the specific TradeStation bar. Every bar on a chart is numbered in ascending order, starting from the left of the chart at 1. When checking for crossovers, you want to start checking on bar number 2 not bar number 1. This is because to check for cross overs, TradeStation looks at the mathematical values of the averages on the bar before and on the current bar. On bar 1, the previous values of the averages are 0 and therefore TradeStation will assume a cross over on bar 1 when none exists.*

This system is designed NOT to pyramid, that is, it does not allow multiple entries in the same direction. Therefore, we also check to see if the variable MP is not equal to zero. When it is not equal to zero, that is, were in either a long or a short position, the system will set the variable LastOrderbar to -999 instead. As explained next, this causes the system not to place any trades until we are in a flat position again:

```
If MP <> 0 then LastOrderBar = -999;
```

We designed EasyLanguage so that when you use the word BUY or SELL, TradeStation will reverse your position.

To exit a long or short position without reversing, you use the words EXITLONG or EXITSHORT, respectively. See the list of Reserved Words in the *EasyLanguage User's Manual*.

We use the UpCh and DnCh variables in our Buy stop and Sell stop orders, respectively. We also use the LastOrderBar variable and CH input to keep these orders active during the last 10 bars (or however many bars are defined in the CH input). Once more than 10 bars have passed since the Delta Average was under zero or crossed over zero, the condition in the first IF-THEN statement evaluates to False and the orders are ignored. Also, when LastOrderBar is equal to -999, the condition is false and the orders are ignored.

```
If BarNumber < LastOrderBar + CH then Begin
        Buy next bar at UpCh Stop;
        Sell next bar at DnCh Stop;
        If MP <> 1 then ExitLong ("LEntryStop") next bar at Lowest(Low, 6) Stop;
        If MP <> -1 then ExitShort ("SEntryStop") next bar at Highest(High, 6) Stop;
End;
```

The Buy stop order is placed at the value set in the UpCh variable, and the Sell stop is placed at the value set in the DnCh variable. We also place a long exit order at the lowest Low of the last 6 bars, and a short exit order at the highest High of the last 6 bars, as discussed under the next section, titled, "Exit Orders."

Exit Orders

This system uses several different exits. The first exit orders are placed on the bar of entry, which is why we make sure we are not in already in a long or short position when we place these first ExitLong and ExitShort orders.

We will exit any long position if the lowest Low of the last 6 bars is hit, and we will exit any short position if the highest High of the last 6 bars is reached. The following instructions place the BUY and SELL stop orders, as discussed previously, and they also place the ExitLong and ExitShort orders.

```
If BarNumber < LastOrderBar + CH then Begin
        Buy next bar at UpCh Stop;
        Sell next bar at DnCh Stop;
        If MP <> 1 then ExitLong ("LEntryStop") next bar at Lowest(Low,6) Stop;
        If MP <> -1 then ExitShort ("SEntryStop") next bar at Highest(High,6) Stop;
End;
```

On the first bar after which a long position is established, which we know by making sure that MP is now equal to one whereas on the previous bar it was not, we calculate a new stop price and store the value in the variable StopPrice. We calculate the new stop price by taking the Low of the current bar minus a 4-bar average of the range:

```
If MP = 1 and MP[1] <> 1 then Begin
        LastOrderBar = -999;
        StopPrice = Low - Average(Range, 4);
End;
```

Then, after the first bar within a long position, we place an order to exit at this stop price or anything lower. On subsequent bars, the stop price is calculated as the current stop price plus a third of the difference between the Low and the current stop price.

```
If MP = 1 then Begin
        Exitlong ("Paralong") next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

Likewise, on the first bar after which a short position is established, we calculate a new stop price and store the value in the variable StopPrice. We calculate the new stop price by taking the High of the current bar plus a 4-bar average of the range.

```
If MP = -1 and MP[1] <> -1 then Begin
        StopPrice = High + Average(Range,4);
        LastOrderBar = -999;
End;
```

Then, after the first bar within a short position, we place an order to exit at this stop price or anything higher. On subsequent bars, the stop price is calculated as the current stop price minus a third of the difference between the current stop price and the High.

```
If MP = -1 then Begin
        Exitshort ("Parashort") next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We did not include margin because we designed this system for stocks, and we specified 200 as the default number of contracts to trade per order.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. In this system, this is determined by the* ***Default Contracts*** *option on this tab.*

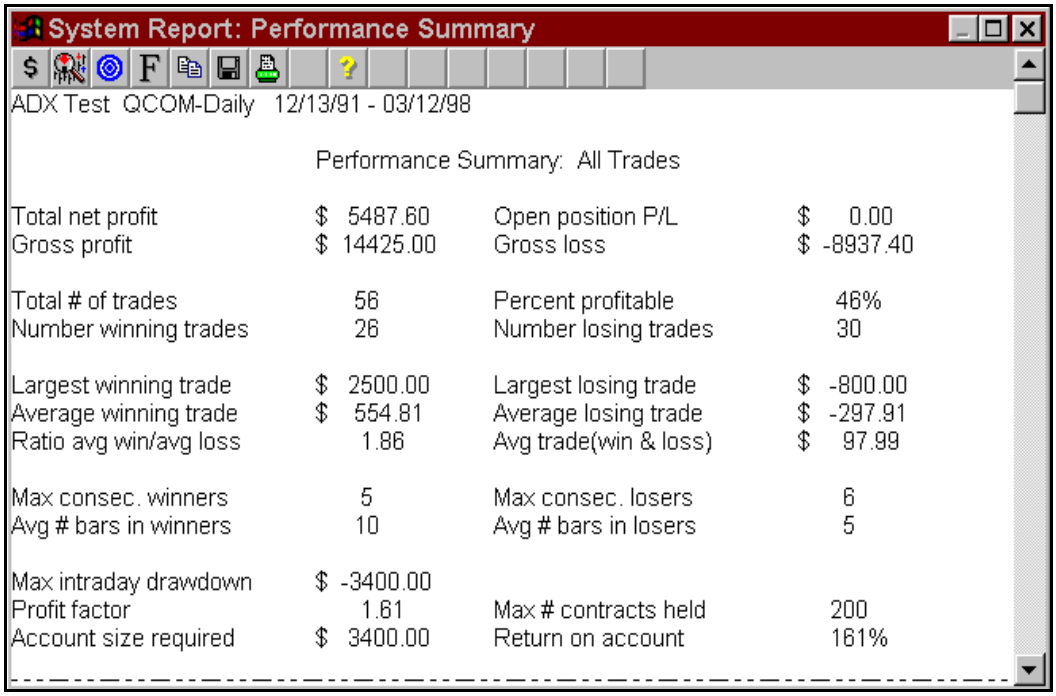
b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position. The default we used is $1,200.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 3 shows a sample System Report for the ADX Differential system applied to a daily chart of Qualcomm. This System Report shows the characteristics typical of trending systems; for example, percent of wining trades is under 50% yet the average win/loss ratio is close to 2.



System Report: Performance Summary

ADX Test QCOM-Daily 12/13/91 - 03/12/98

Performance Summary: All Trades

Total net profit	$ 5487.60	Open position P/L	$ 0.00
Gross profit	$ 14425.00	Gross loss	$ -8937.40
Total # of trades	56	Percent profitable	46%
Number winning trades	26	Number losing trades	30
Largest winning trade	$ 2500.00	Largest losing trade	$ -800.00
Average winning trade	$ 554.81	Average losing trade	$ -297.91
Ratio avg win/avg loss	1.86	Avg trade(win & loss)	$ 97.99
Max consec. winners	5	Max consec. losers	6
Avg # bars in winners	10	Avg # bars in losers	5
Max intraday drawdown	$ -3400.00		
Profit factor	1.61	Max # contracts held	200
Account size required	$ 3400.00	Return on account	161%

Figure 3. Sample System Report for the ADX Differential System

When evaluating any system, make sure you look at the largest winning trade and what percentage of the net profit this trade represents. Because trending systems are designed to capture the big moves, the percentage of the net profit represented by the largest winning trade will be greater than another type of system, say a Support and Resistance system. However, all the profit coming from one trade is not desirable. Ideally, you want part of the profit coming from a few big trades, not a single big trade. This report shows nearly half of the profits coming from one trade; we should definitely evaluate this system on additional data to determine whether or not these are statistical outliers.

Suggestions for Improvement

When determining the entry price, we wait for the 4-bar exponential average of the ADX to be under zero or to cross over zero to place orders at the close plus or minus half of the 4-bar average of the range.

However, when this 4-bar exponential average of the ADX crosses over zero, the underlying symbol has already started to trend; therefore, one way we can try to improve this system is by trying to enter the market sooner. To do this, we can try to place tighter entry prices.

Head and Shoulders

Head and Shoulders is one of the classic bar patterns that have been followed by traders throughout the years. It is a trend reversal pattern, meaning that it is used to determine when trends are reversing. One of the first publications describing them in any detail is the book by Robert D. Edwards and John Magee, titled *Technical Analysis of Stock Trends* (7th Edition, John Magee Inc., Chicago, IL), which uses the following terms to describe the Head and Shoulder Top pattern. Please refer to Figure 4 as you read this description.

- A strong rally, climaxing a more or less extensive advance (Point A).
- Followed by a minor recession on which the volume runs considerably less than it did during the days of the rally. At this point we have the left shoulder (Point D).
- Another higher volume advance which reaches a higher level than the top of the left shoulder (Point B).

- A second decline that reaches a level near the bottom between the shoulder and the previous peak (Point E). At this point, we have completed the head of the pattern.
- A third rally with less volume than the previous two rallies and which fails to reach prices of the head, which is the second rally (Point C).
- Finally, a decline, which crosses under the neckline (a line drawn from Point D to E).

Figure 4 illustrates the descriptions used by Edward and Magee to define a Head and Shoulders Top formation.

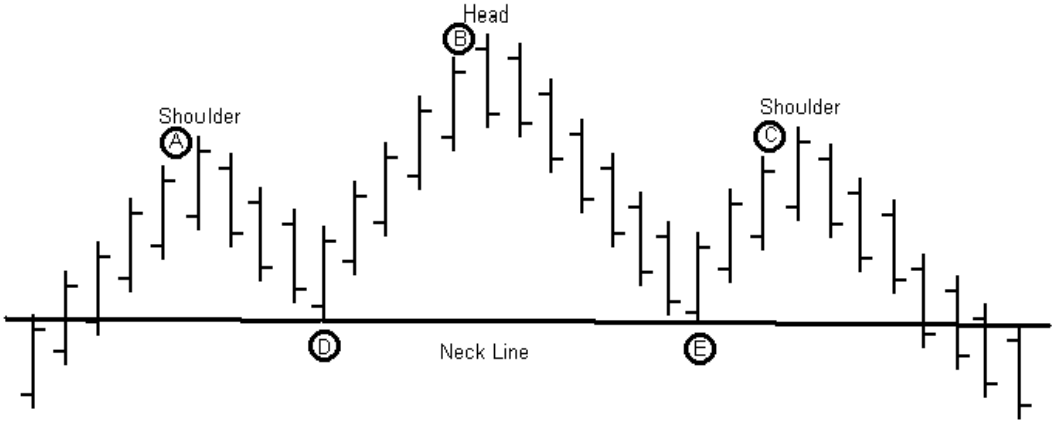


Figure 4. Illustration of a classic Head and Shoulders Top bar pattern.

The exact inverse is true for a Head and Shoulders Bottom pattern. Please refer to Figure 5 when reading the descriptions:

- A decline, climaxing a more or less extensive decline (Point A).
- A minor recovery on which the volume runs less than it did during the final bars of the decline (Point D), representing the shoulder.
- Another decline which carries prices below the bottom of the previous shoulder (Point B).
- A second rally that carries prices near the first rally (Point E).
- A third decline with decidedly less volume than the left shoulder or the head (Point C).
- A final rally that crosses over the neck line.

Figure 5 illustrates the descriptions used by Edward and Magee to define a Head and Shoulders Bottom formation.

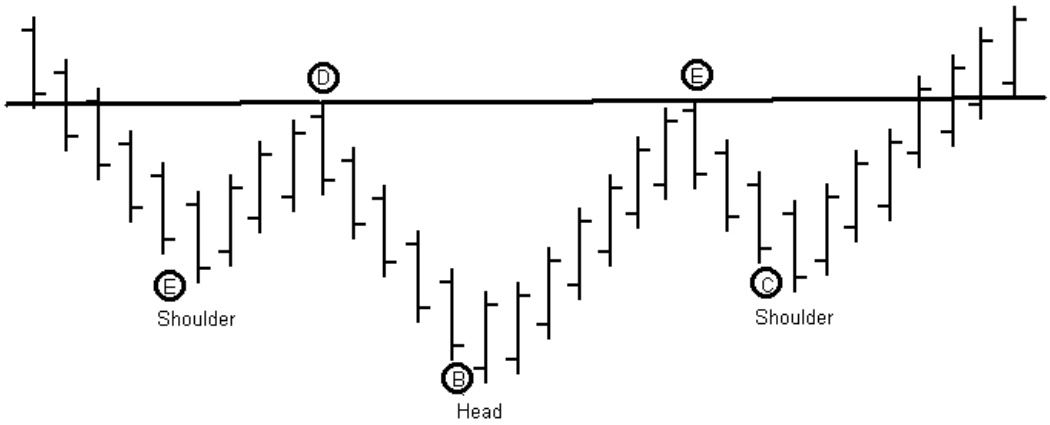


Figure 5. Illustration of a classic Head and Shoulders Bottom bar pattern

We decided to create a trading system based on both the Head and Shoulders Top and Bottom patterns. We will sell short whenever the price crosses under the neckline of a Head and Shoulder Top pattern, and we will buy whenever the price crosses under the neckline of a Head and Shoulder Bottom pattern.

Once we decided upon our entries, we determined how we were going to exit the market. On the first bar after we enter our long position, we will define an arbitrary point below the Low of our entry bar. We will subtract the average of the range of the last 4 bars from the Low of the bar of entry and use that as our exit point for the first bar. We can always modify this at a later time to place a tighter stop. For our exits on the remaining bars, we will add a third of the distance between the low of the bar and the previous exit point to determine our next day's exit point.

We will do the exact opposite for short positions. On the first bar after we enter our short position, we will add a 4-bar average of the range to the High of the bar of entry and use this as a stop for the first bar. Thereafter, we will subtract a third of the distance from the previous stop and the High of the bar from the previous stop point to determine next bar's exit price.

Also, we will draw trendlines and text on the chart when we find the bar patterns.

Defining your Trading Rules

In this system, we defined long and short entries as well as exit orders. The long and short entries reverse your position, whereas the exits close out your existing position and exit you from the market. We also performed significant setup work, which involved identifying the Head and Shoulders patterns and necklines. In addition, we drew trendlines and text for visual analysis. The setup, entry and exits are described next.

Setup

A significant portion of this trading system involves setup. In order to identify the Head and Shoulders patterns, we used swing high and swing low patterns.

A **swing high** bar (with a default strength of 2) is a bar whose high is higher than the 2 bars to its left and at least as high as the two bars to its right.

Similarly, a **swing low** bar (with a default strength of 2) is a bar whose low is lower than the 2 bars to its left and at least as low as the 2 bars on its right.

Long Entries

a) Enter a long position when the neckline of a Head and Shoulder Bottom pattern formation is broken.

Long Entries

a) Enter a short position when the neckline of a Head and Shoulder Top pattern formation is broken.

Exits

a) Once the bar of entry of a long position has closed, calculate the 4-bar average of the range and divide it by 4. Then, subtract this value from the low. This will be our long exit price for bar 2.

b) From the second bar, get the exit price calculated in the last point and add to it a third of the difference between the low and the previous stop price. Repeating this operation at the end of every bar will give you the exit price for bar three and beyond.

c) Once the bar of entry of a short position is closed, calculate the 4-bar average of the range and divide it by 4. Then add this value to the high. This will be your short exit price for bar 2.

d) From the second bar, get the exit price calculated in the last point and add to it a third of the difference between the low and the previous stop price. Repeating this operation at the end of every bar will give you the exit price for bar three and beyond.

e) Place a money management protective stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts are being traded.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

We created this system in modules. We wrote the criteria for the Head and Shoulders Top bar pattern in one system (STAD2: H&S Top) and the criteria to follow the bottom Head and Shoulders Bottom bar pattern in a second system (STAD2: H&S Bot). Then we created a third system that references the first and second system (STAD2: H&S). *You only need to apply the one system, STAD2: H&S.*

In order to reference one system from another, we used the 'IncludeSystem' EasyLanguage command. This command is very useful and commonly used. There are two main advantages to using the 'IncludeSystem' command:

a) In TradeStation 4, each system can be a maximum of 64K in size. If your system is large, it is possible to exceed this limit. One easy way to overcome this limit is to break your system into one or more systems and then combine them using the 'IncludeSystem' command. Then, each system has 64K allocated to it.

b) You can use the 'IncludeSystem' command to include instructions that you use often. For example, if there is an exit you use often, you can write it into a system and then include it any time you want to use it in a system you are writing.

Following are the instructions for the main system. Notice how short it is. Notice also that we defined inputs. You don't have to, however, if you want to be able to modify the systems you included, you have to define the inputs in the main system.

EasyLanguage Instructions: Head and Shoulders System (STAD2: H&S)

```
Input: TopStren(5), BotStren(5), TLDraw(True), Color1(Tool_Yellow),
        Color2(Tool_Cyan);

IncludeSystem:"STAD2: H&S Bot",TOPSTREN,BOTSTREN,TLDRAW,COLOR1,COLOR2;
IncludeSystem:"STAD2: H&S Top",TOPSTREN,BOTSTREN,TLDRAW,COLOR1,COLOR2;
```

This system merges the two systems and converts them into one, so they can be applied as one system on any given chart. The EasyLanguage instructions for the Head and Shoulders Top system are listed next. The instructions for the Head and Shoulders Bottom system are not listed because they exactly mirror the H&S Top instructions. The subsequent discussion covers the Head and Shoulders Top system, however, the explanations can be applied to the Head and Shoulders Bottom system, and of course, the final main system is the two together.

IMPORTANT NOTE: *The STAD2: H&S system on your CD contains an error in the inputs. The correct version is the version shown above. The system on your CD will read as follows (the bold items are the incorrect items):*

```
Input: TopStren(5), BotStren(5), ShowMsg(True), ShowTL(True), Color1(Tool_Yellow),
        Color2(Tool_Cyan);

IncludeSystem:"STAD2: H&S Bot",TOPSTREN,BOTSTREN,SHOWTL,COLOR1,COLOR2;
IncludeSystem:"STAD2: H&S Top",TOPSTREN,BOTSTREN,SHOWTL,COLOR1,COLOR2;
```

Please use the PowerEditor to correct these items before plotting the system.

EasyLanguage Instructions: Head and Shoulders Top System (STAD2: H&S Top)

```
Inputs: TopStren(5), BotStren(5), TLDraw(True), Color1(Tool_Yellow), Color2(Tool_Cyan);
Array: SwingHighs[3,5](0), SwingLows[3,3](0);
Vars: TextColor(Color2), TL_ID(-1), LastCrossedTL(-1), MP(0), StopPrice(0);
```

```
{Fills SwingHighs array when Swing High bars encountered}
If SwingHighBar(1, High, TopStren, TopStren + 1) = TopStren then Begin
        for Value1 = 2 downto 0 Begin
                SwingHighs[Value1+1,0] = SwingHighs[Value1,0];
                SwingHighs[Value1+1,1] = SwingHighs[Value1,1];
                SwingHighs[Value1+1,2] = SwingHighs[Value1,2];
                SwingHighs[Value1+1,3] = SwingHighs[Value1,3];
                SwingHighs[Value1+1,4] = SwingHighs[Value1,4];
                SwingHighs[Value1+1,5] = SwingHighs[Value1,5];
        End;
        SwingHighs[0,0] = BarNumber[TopStren];
        SwingHighs[0,1] = Date[TopStren];
        SwingHighs[0,2] = Time[TopStren];
        SwingHighs[0,3] = High[TopStren];
        SwingHighs[0,4] = LinearRegAngle( Volume, TopStren * 2 );
        SwingHighs[0,5] = 0;
End;
{Fills SwingLows array when Swing Low bars encountered}
If SwingLowBar(1, Low, BotStren, BotStren+1) = BotStren then Begin
        for Value1=2 downto 0 Begin
                SwingLows[Value1+1,0] = SwingLows[Value1,0];
                SwingLows[Value1+1,1] = SwingLows[Value1,1];
                SwingLows[Value1+1,2] = SwingLows[Value1,2];
                SwingLows[Value1+1,3] = SwingLows[Value1,3];
        End;
                SwingLows[0,0] = BarNumber[BotStren];
                SwingLows[0,1] = Date[BotStren];
                SwingLows[0,2] = Time[BotStren];
                SwingLows[0,3] = Low[BotStren];
End;
{Specifies comparisons to be made between Swing High values stored in the SwingHighs array}
Condition1 = SwingHighs[0,0] > SwingLows[0,0] AND SwingLows[0,0] > SwingHighs[1,0] AND
        SwingHighs[1,0] > SwingLows[1,0] AND SwingLows[1,0] > SwingHighs[2,0];
Condition2 = SwingHighs[1,3] > SwingHighs[2,3] AND SwingHighs[1,3] > SwingHighs[0,3];
Condition3 = SwingLows[0,3] < SwingHighs[2,3] AND SwingLows[1,3] < SwingHighs[0,3];
If SwingHighs[1,3] - MinList(SwingLows[0,3], SwingLows[1,3]) <> 0 then
        Condition4 = (AbsValue(SwingLows[0,3] - SwingLows[1,3])) / (SwingHighs[1,3] -
                MinList(SwingLows[0,3], SwingLows[1,3]))  < .1 else Condition4 = True;
{Evaluates comparisons to determine if Head & Shoulders Top pattern is present, and if so, draws
trendline and text}
If Condition1 AND Condition2 AND Condition3 AND Condition4 AND SwingHighs[0,5] = 0 then Begin
        If TextColor = Color2 then TextColor = Color1 else TextColor = Color2;
        SwingHighs[0,5] = 1;
        Value99 = Text_New( SwingHighs[0,1], SwingHighs[0,2], SwingHighs[0,3], "Shoulder" );
        Value98 = Text_SetStyle(Value99,2,1);
        Value98 = Text_SetColor( Value99 , TextColor );
        Value99 = Text_New( SwingHighs[1,1], SwingHighs[1,2], SwingHighs[1,3], "Head" );
        Value98 = Text_SetStyle(value99,2,1);
        Value98 = Text_SetColor( value99 , TextColor );
        Value99 = Text_New( SwingHighs[2,1], SwingHighs[2,2], SwingHighs[2,3], "Shoulder" );
```

```
            Value98 = Text_SetStyle(value99,2,1);
            Value98 = Text_SetColor( value99 , TextColor );
            If TLDraw then Begin
                  TL_ID = TL_New( SwingLows[1,1], SwingLows[1,2], SwingLows[1,3],
                        SwingLows[0,1], SwingLows[0,2], SwingLows[0,3] );
                  Value99 = TL_SetColor( TL_ID, TextColor );
            End;
End;
```

{Determines if trendline drawn and whether it's valid to use, then compares close price to trendline to find penetration}

```
If TL_ID >= 0 AND LastCrossedTL <> TL_ID then
      If Close crosses under TL_GetValue( TL_ID, Date, Time ) then Begin
            Sell next bar at Open;
            LastCrossedTL = TL_ID;
      End;
```

{Calculates stop price and places exit orders}

```
MP = MarketPosition;
If MP = -1 and MP[1] <> -1 then StopPrice = High +  Average(Range,4);
If MP = -1 then Begin
      Exitshort ("parashort") next bar at StopPrice stop;
      StopPrice = StopPrice - (.3 * (StopPrice - High));
      Value99 = Text_New(Date,Time,StopPrice,".");
End;
```

Inputs

Following is the list of all the inputs we used in the Head and Shoulders Top System:

Input	Default Value	Description
TopStren	5	Minimum number of bars to the left and right of the top of the rallies to be considered significant
BotStren	5	Minimum number of bars to the left and right of bottoms of declines to be considered significant
TLDraw	True	True/false expression that will enable or disable the drawing of the neck trend line
Color1	Tool_Yellow	Color of the odd numbered patterns
Color2	Tool_Cyan	Color of the even numbered patterns

For this system, we define the following variables:

```
Vars: TextColor(Color2), TL_ID(-1), LastCrossedTL(-1), MP(0), StopPrice(0);
```

We also make use of arrays, as discussed next.

Setup

The setup for this system consists of identifying the Head and Shoulders patterns and storing the relevant information for comparison, which is accomplished using arrays.

Arrays

Arrays are described in more detail in Chapter 2 of the previous STAD Club volume, Volume 1.

In order to find the Points A, B, C, D and E, we'll use arrays. Looking back at the chart of the Head and Shoulder Top formation, we can see that Points A, B and C are swing highs and Points C and D are swing lows. Therefore, what we will do to find these patterns is keep the information of the last 3 swing highs and the last 3 swing lows in two different arrays. We'll call the array for the swing highs SwingHighs and the array for the swing lows SwingLows. The Array Declaration statement is:

```
Array: SwingHighs[3,5](0), SwingLows[3,3](0);
```

A *switch* is a mechanism used to monitor whether an incident has occurred. You can use a variable—in this case an element in an array—whose value you change once an incident has occurred. Then, you can execute certain EasyLanguage instructions or not depending on the value in the variable or element.

The SwingHighs has 4 rows (0, 1, 2 and 3) and 6 columns (0, 1, 2, 3, 4, 5) and the SwingLows has 4 rows and 4 columns. The arrays hold information of the top and bottom bars of the swings. Specifically, they will hold the bar number, the date, the time, and the price (of the high or the low). The SwingHighs array holds two additional pieces of information, a switch and a volume calculation. Whenever the swing high is used to draw a pattern, it will be marked as used by means of this switch. This will avoid drawing the same pattern twice.

Note: *In the STAD2: H&S Bottom system, the opposite is true; the SwingLows array has two additional pieces of information. Also, even though we are using one column to store the linear regression angle of the volume, we do not actually use this piece of information. We were thinking of including a volume criteria in our system, as suggested by Edwards and Magee, but didn't. We discuss this in detail in the section titled, "Suggestions for Improvement."*

Every time a new swing high is found, we will need to move all the rest of the elements down one row, drop the oldest swing high and add the new swing high to the top of the array. We store the bar number, date, time, high and linear regression angle of the volume for each swing high:

```
If SwingHighBar(1, High, TopStren, TopStren + 1) = TopStren then Begin
        for Value1=2 downto 0 Begin
                SwingHighs[Value1+1,0] = SwingHighs[Value1,0];
                SwingHighs[Value1+1,1] = SwingHighs[Value1,1];
                SwingHighs[Value1+1,2] = SwingHighs[Value1,2];
                SwingHighs[Value1+1,3] = SwingHighs[Value1,3];
                SwingHighs[Value1+1,4] = SwingHighs[Value1,4];
                SwingHighs[Value1+1,5] = SwingHighs[Value1,5];
        End;
        SwingHighs[0,0] = BarNumber[TopStren];
        SwingHighs[0,1] = Date[TopStren];
        SwingHighs[0,2] = Time[TopStren];
        SwingHighs[0,3] = High[TopStren];
        SwingHighs[0,4] = LinearRegAngle( Volume, TopStren * 2 );
        SwingHighs[0,5] = 0;
End;
```

The same procedure is used for the swing lows. When a new swing low is found, all the old swing lows are moved one row down and then the new swing low is added to the top of the array. For the swing lows, we store the bar number, the date, the time and the low:

```
If SwingLowBar(1, Low, BotStren, BotStren+1) = BotStren then Begin
        for Value1=2 downto 0 Begin
                SwingLows[Value1+1,0] = SwingLows[Value1,0];
                SwingLows[Value1+1,1] = SwingLows[Value1,1];
                SwingLows[Value1+1,2] = SwingLows[Value1,2];
                SwingLows[Value1+1,3] = SwingLows[Value1,3];
        End;
        SwingLows[0,0] = BarNumber[BotStren];
        SwingLows[0,1] = Date[BotStren];
        SwingLows[0,2] = Time[BotStren];
        SwingLows[0,3] = Low[BotStren];
End;
```

Once we have our significant swing points stored in our arrays, we need to define the conditions that will check for a combination of swing points that make up a Head and Shoulder Top formation. First, we need to determine the sequence of the swing points. The last swing high encountered (Point C) has to be more recent than the most recent swing low (Point E). The most recent swing low (Point E) has to be more recent than the second most recent swing high (Point B). The second most recent swing high (Point B) has to be newer than the second most recent swing low (Point D). Finally, the second most recent swing low (Point D) has to have a greater bar number than the third swing high (Point A). Said another way, the points have to be defined in the following order, from oldest to newest:

A -> D -> B -> E -> C

The EasyLanguage instructions to check for the pattern are:

```
Condition1 = SwingHighs[0,0] > SwingLows[0,0] AND SwingLows[0,0] > SwingHighs[1,0] AND
        SwingHighs[1,0] > SwingLows[1,0] AND SwingLows[1,0] > SwingHighs[2,0];
```

Also, Point B in the chart needs to be higher than points A and C. Point D in the chart needs to be lower than Point C and Point E has to be lower than Point A. Finally, the distances from Point B to D and from Point B to E have to be within 10% of each other. This will make the neck line be nearly horizontal.

The EasyLanguage instructions to check for the pattern are:

```
Condition2 = SwingHighs[1,3] > SwingHighs[2,3] AND SwingHighs[1,3] > SwingHighs[0,3];

Condition3 = SwingLows[0,3] < SwingHighs[2,3] AND SwingLows[1,3] < SwingHighs[0,3];

If SwingHighs[1,3] - MinList(SwingLows[0,3], SwingLows[1,3]) <> 0 then
        Condition4 = (AbsValue(SwingLows[0,3] - SwingLows[1,3])) / (SwingHighs[1,3] -
                MinList(SwingLows[0,3], SwingLows[1,3]))  < .1 else Condition4 = True;
```

If the conditions defined above are all true, this means that we have found most of the bar pattern that conforms a Head and Shoulder Top. Now, if the switch cell of the SwingHighs array is equal to 0 (zero), this means that we have not drawn this pattern already so we will do so. We will place drawing objects on the chart that represent our pattern, alternating between two colors when writing the Head and Shoulder messages and drawing the trend line in the chart:

```
If Condition1 AND Condition2 AND Condition3 AND Condition4 AND SwingHighs[0,5] = 0
then Begin
        If TextColor = Color2 then TextColor = Color1 else TextColor = Color2;
```

Because we are drawing this pattern, we will change the switch cell in our SwingHighs array to 1. We will write the word ‘Shoulder’ over Points A and C of our chart, and the word ‘Head’ over Point B of our chart.

```
SwingHighs[0,5] = 1;
Value99 = Text_New(SwingHighs[0,1], SwingHighs[0,2], SwingHighs[0,3],
          "Shoulder");
Value98 = Text_SetStyle(Value99,2,1);
Value98 = Text_SetColor(Value99 , TextColor );
Value99 = Text_New( SwingHighs[1,1], SwingHighs[1,2], SwingHighs[1,3], "Head");
Value98 = Text_SetStyle(Value99,2,1);
Value98 = Text_SetColor(Value99 , TextColor );
Value99 = Text_New( SwingHighs[2,1], SwingHighs[2,2], SwingHighs[2,3],
          "Shoulder");
Value98 = Text_SetStyle(Value99,2,1);
Value98 = Text_SetColor(Value99 , TextColor );
```

If the input TL_Draw is true, then we will draw a trend line from Point D to Point E on our chart.

```
        If TLDraw then Begin
                TL_ID = TL_New( SwingLows[1,1], SwingLows[1,2], SwingLows[1,3],
                        SwingLows[0,1], SwingLows[0,2], SwingLows[0,3] );
                Value99 = TL_SetColor( TL_ID, TextColor );
        End;
End;
```

Long Entries

When you draw a trendline, you use the TL_New function, which you assign to a numeric variable. When the trendline is drawn successfully, the function returns a positive integer that represents the trendline ID number.

When the trendline is not drawn, the function returns a negative integer that identifies the reason it was not drawn. The error codes that apply to trendlines are:

- **-2** The object identifier was invalid
- **-3** The data number (Data1, etc.) passed to the function was invalid
- **-5** The beginning and ending points were the same
- **-6** The function was unable to load the default values for the tool
- **-7** Unable to add the object; possibly due to an out of memory condition
- **-8** Invalid pointer
- **-9** Previous failure
- **-10** Too many trendlines on the chart

First we check to make sure the trendline ID is greater than or equal to zero. If the TL_ID is less than zero, this would mean that either no trendlines are drawn or there was a problem drawing a trendline (when the TL_New() instruction cannot successfully draw a trend line it will return a negative number that indicates why it could not draw the trend line).

Then, if the Close crosses under the value of the trend line for the current bar, we will sell short next bar at the open. We will also store the number of the trendline that generated the order in the variable LastCrossedTL. This variable will be used to avoid having the same trendline generate more than one entry order

```
        If TL_ID >= 0 AND LastCrossedTL <> TL_ID then
                If Close crosses under TL_GetValue( TL_ID, Date, Time ) then Begin
                        Sell next bar at Open;
                        LastCrossedTL = TL_ID;
                End;
```

Exit Orders

At the first bar of a short position, the system will define StopPrice as the low of the current bar minus a 4-bar average of the range.

```
        If MP = -1 and MP[1] <> -1 then StopPrice = High +  Average(Range,4);
```

Then, we will then place an order to exit a short position at the stop price or anything lower. Once this is done, the stop price for the next bar is calculated as the current stop price plus a third of the difference between the Low and the current stop price. We also draw text on the chart at the level of the stop price (in this case we're drawing a period).

```
        If MP = -1 then Begin
                Exitshort ("parashort") next bar at StopPrice stop;
                StopPrice = StopPrice - (.3 * (StopPrice-high));
                Value99 = Text_New(Date,Time,StopPrice,".");
        End;
```

We use the period strictly as a visual aid. It serves to show where the stop is for the next bar. Sometimes we'll use an X to make the point more visible, or we'll eliminate it altogether. The text is not necessary for the system to function, it is simply for visual effect.

The long side of the system is the mirror of these instructions.

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We did not include margin because we designed this system for stocks, and we left the default number of contracts to trade per order at 1.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number*

of shares the system is buying and selling. In this system, this is determined by the ***Default Contracts*** *option on this tab.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position. The default we used is $1,200.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 6 shows a sample System Report for the Head and Shoulders system. Remember, we used a modular approach to create this system, and the system we apply is the main system, STAD2: H&S. The other two, STAD2: H&S Top and STAD2: H&S Bottom, are referenced by this main system.

Head and Shoulder patterns are very rare, and it will be hard to find a chart that contains a significant number of these patterns. Therefore, it will be hard to determine if this system is consistently profitable. The system report in Figure x shows 12 trades of which 5 were winning trades, and it shows a 3.2 ratio over the 7 losers, which is typical for trending patterns. As encouraged as we are by this idea, this system (and these patterns) should be applied to many more symbols (and different data compressions) to confirm the reliability and profitability of this trading idea.

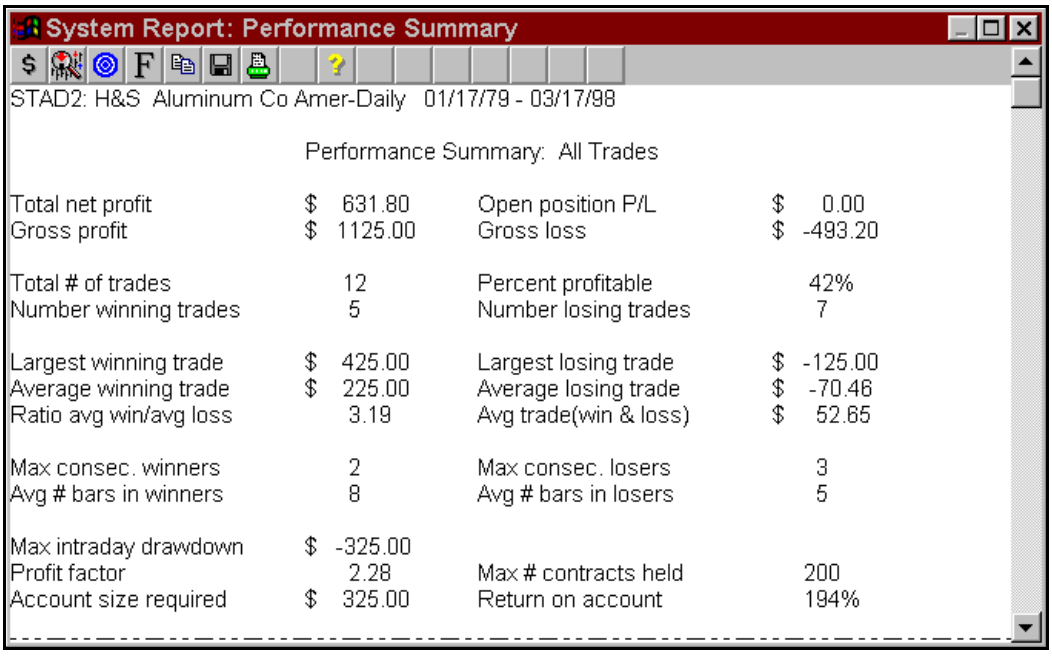


Figure 6. Sample system report for the Head and Shoulders system.

Suggestions for Improvement

The effect of a head and shoulder pattern will not last forever, we should only pay attention when the neck line of a recent head and shoulder is broken. As the systems currently stand, it looks for the price to break the trend line indefinitely or until a new pattern is detected.

Volume is an important part of the bar pattern as originally described by Edwards and Magee. The system as we have written it does not include the analysis of volume, however, since we are already storing the results of the calculation of a linear regression angle of the volume, we could easily incorporate this into our analysis at a later date.

On Balance Volume: New Highs, New Lows

This system is similar to the Advance Decline Divergence system. It is based on the same premise. However, the Advance Decline Divergence system looks for a divergence between the price movement and the volume, rather than just new highs and lows in volume, in an effort to identify tops and bottoms. Since the AD-Divergence system is looking for tops and bottoms, it is considered a Support and Resistance system.

One of the most accepted trading principles is that an increase in price movement will be accompanied by an increase in volume. Therefore, it is generally accepted that in order for a rally to evolve into a trend, it must be accompanied by an increase in volume. We decided to look into this concept further to see if we could build a trending system based on it.

We know that On Balance Volume (OBV) is a standard indicator that shows the relationship between price movement and the volume that originates the price movement. It is quite simple in its calculation: if the close of the current bar is greater than the close of the previous bar, then you add the volume of the current bar to a running total; conversely, if the close of the current bar is less than the close of the previous bar, then you subtract the volume from the running total; and finally, if the price is unchanged, then the running total is left unchanged as well. The running total is referred to as the on balance volume.

We began searching for instances when the OBV Indicator made new highs and lows. We arbitrarily chose the last 15 bars as the period in which to look for these new highs and lows. To find these instances, we wrote a new ShowMe Study, called OBV New H/L, and applied it to different charts. Figure 7 shows a chart to which we've applied the first version of the OBV New H/L ShowMe Study. Notice the dots over the bars where the OBV made a new 15-bar high and the dots under the bars where the OBV made a new 15-bar low. We have also applied the standard On Balance Volume Indicator.

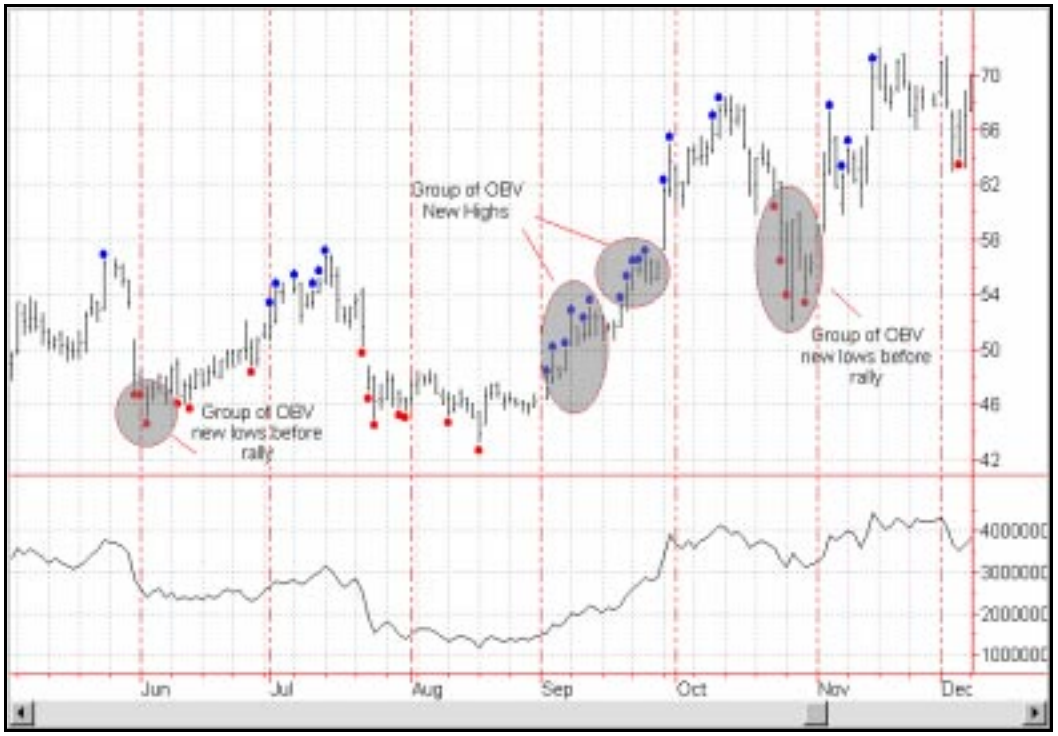


Figure 7. The OBV New H/L ShowMe study applied to a chart.

It appears from looking at the chart that whenever there are new OBV highs or lows, the price tends to rally up. Also, it appears that new highs and new lows in OBV tend to group together before pronounced rallies. This was encouraging. However, we also noticed that new highs and lows in the on balance volume tend to occur frequently, a little too frequently to base a system on this criteria, so we decided to find the bars where the OBV has made several new highs and lows within a short period of time, and in this way focus in on only the pronounced rallies.

Therefore, we rewrote the OBV New H/L ShowMe study to include inputs so we could control how may new highs and new lows had to occur in a specified number of bars. After enough testing, we decided that 5 new highs in the last 6 bars and 3 new lows in the last 6 bars are significant.

Figure 8 shows the same ShowMe study, modified as described to the same chart.

Figure 8. The OBV New H/L ShowMe Study and System applied to a chart.

The results were definitely encouraging, so we went ahead and used this as our long entry criteria; we decided to include only long entries at this time.

Once we had decided upon our entry criteria, we turned our attention to exits. On the first bar after we enter our long position, we will define an arbitrary point below the Low of our entry bar. We will subtract the average of the range of the last 4 bars from the Low of the bar of entry and use that as our exit point for the first bar. We can always modify this at a later time to place a tighter stop. For our exits on the remaining bars, we will add a third of the distance between the low of the bar and the previous exit point to determine our next day's exit point.

Figure x above also shows the resulting On Balance Volume: New Highs, New Lows system applied.

Defining your Trading Rules

In this system, we defined only long entries and exit orders. There is some simple setup work that involves looking for new highs and new lows in the OBV. The setup, entries and exits are as follows:

Setup

a) Compare the OBV of the current bar to the highest OBV of the last 15 bars and to the lowest OBV of the last 15 bars.

Long Entries

a) If the OBV has made 5 new highs in the last 6 bars then buy 100 contracts at the open of the next bar.

b) If the OBV has made 3 new lows in the last 6 bars, then buy 200 contracts at the open of the next bar.

Exits

a) Once the bar of entry is closed, we calculate the 4-bar average of the range, divide the average by 4, and then subtract this value from the Low of the current bar. This is our long exit price for bar 2.

b) For all bars after bar 2, we obtain the previously calculated exit price and add to it a third of the difference between the Low and the previous stop price. We repeat this operation at the end of every bar to give us the exit price for all subsequent bars.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: On Balance Volume: New Highs, New Lows System (STAD2: OBV New Highs)

```
Input: Length(15), BarsBack(6), NewHighs(5), NewLows(3);

Vars: OBVNewHigh(false), MP(0), StopPrice(0), OBVNewLow(False);

{Finds new highs and lows in OBV}
OBVNewHigh = OBV >= Highest( OBV, Length );

OBVNewLow = OBV <= Lowest( OBV, Length );

{Determines if the specified number of new highs and new lows have occurred in the specified
number of bars and places buy orders}
If MRO( OBVNewHigh , BarsBack, NewHighs ) > -1 then
        Buy ("New High") 100 shares next bar at market;

If MRO( OBVNewLow , BarsBack, NewLows ) > -1 then
        Buy ("New Low") 200 shares next bar at market;

{Calculates stop price and places exit order}
MP = MarketPosition;

If MP = 1 and MP[1] <> 1 then
        StopPrice = low - Average(Range,4);

If MP = 1 then Begin
        Exitlong ("ExitTrade") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low -StopPrice)/3;
End;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default Value	Description
Length	15	Number of bars to use when looking for new highs/lows
BarsBack	6	Number of bars to look back when looking for groups of new highs/lows
NewHighs	5	Number of new highs in the last BarsBack for the system to enter a position
NewLows	3	Number of new lows in the last BarsBack for the system to enter a position

Additionally, we define the following variables:

```
Vars: OBVNewHigh(False), MP(0), StopPrice(0), OBVNewLow(False);
```

Setup

We will look for new highs and new lows in the OBV and assign the result of these conditions into the variables OBVNewHigh and OBVNewLow. The OBV function calculates and returns the on balance volume.

```
OBVNewHigh = OBV >= Highest( OBV, Length );
OBVNewLow = OBV <= Lowest( OBV, Length );
```

Long Entries

If in the last 6 bars the OBV has made 5 new highs, then buy 100 shares next bar at market. We use the MRO function to determine whether or not this has occurred.

```
If MRO( OBVNewHigh , BarsBack, NewHighs ) > -1 then
        Buy ("New High") 100 shares next bar at market;
```

If in the last 6 bars the OBV has made 3 new lows, then buy 200 shares next bar at market:

```
If MRO( OBVNewLow , BarsBack, NewLows ) > -1 then
        Buy ("New Low") 200 shares next bar at market;
```

Exit Orders

At the first bar of a long position, the system will define StopPrice as the low of the current bar minus a 4-bar average of the range. We use the MarketPosition function to determine our current position in the market and assign the value to the variable MP:

```
MP = MarketPosition;

If MP = 1 and MP[1] <> 1 then StopPrice = Low - Average(Range,4);
```

If in a long position, then we will place an order to exit at the stop price or anything lower. Once this is done, the stop price for the next bar is calculated as the current stop price plus a third of the difference between the low and the current stop price.

```
If MP = 1 then Begin
        Exitlong ("ExitTrade") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it:

a) In the **Costs** tab, we entered the appropriate amounts for the default number of contracts to trade, commission and slippage. We did not include margin because we designed this system for stocks.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. Unless specified within the system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the option **Allow Multiple Entries in the same direction by same and different entry orders**. This is because this system is designed to scale into a position.

d) Also in the **Properties** tab, we typed the number 4 in the option **Maximum number of entries per position**. This limits the number of times the system can enter the market in the same direction to 4. For example, if we type 4 in this option and the system is buying blocks of 100 shares, the system to hold a maximum of 400 shares at any given time.

Testing and Improving

This system is to an extent a hybrid between a trending and a volatility expansion system. By studying the OBV, the system is attempting to catch the beginning of a trend by following increases in price and volume (increase in volatility). This is why our system results, shown in Figure 9, are somewhat abnormal for a trending system; the percentage of wining trades is around 55% and the ratio average winning versus losing trade is somewhat low at 1.61.

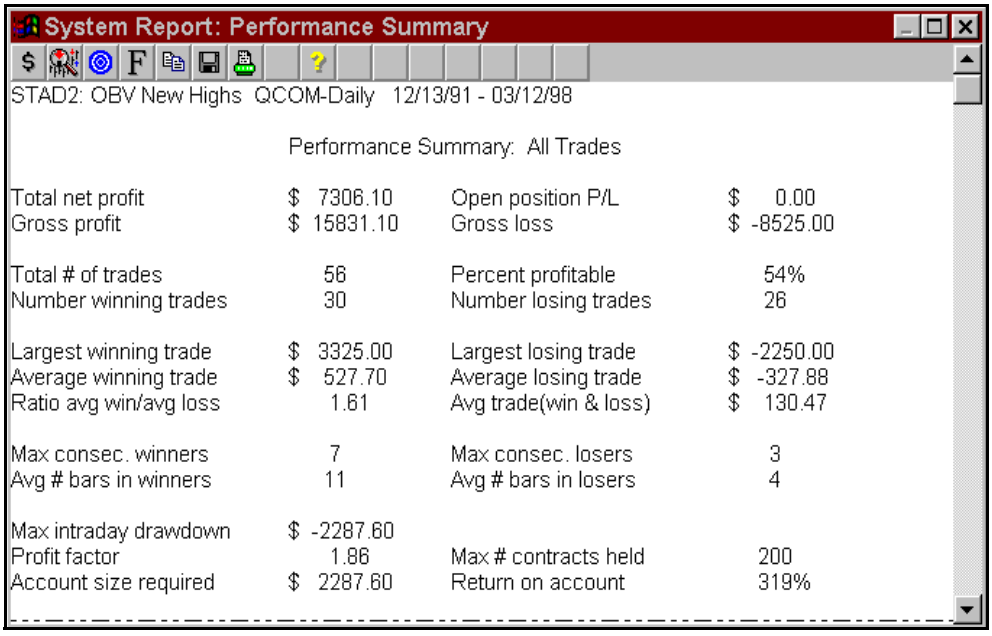


Figure 9. Sample system report for the On Balance Volume: New Highs, New Lows system

When evaluating any system, make sure you look at the largest winning trade and what percentage of the net profit this trade represents. Because trending systems are designed to capture the big moves, the percentage of the net profit represented by the largest winning trade will be greater than another type of system, say a Support and Resistance system. However, all the profit coming from one trade is not desirable. Ideally, you want part of the profit coming from a few big trades, not a single big trade. This report shows nearly half of the profits coming from one trade; we should definitely evaluate this system on additional data to determine whether or not these are statistical outliers.

Suggestions for Improvement

It is important to consider that a new low in OBV can be a bearish sign if the symbol under observation has a overall bearish outlook. In cases of securities with acceptable (positive) fundamental information, this can be considered a sign that the symbol has reached a support level. Dial Data, Inc. (and our Historical data CD) provides historical Earnings Per Share that can be used to determine the performance of the company, this can be used to establish a short position if the company is performing poorly and a new OBV low is hit.

Also, as explained before, price change will always be accompanied with high volume. If we establish a position and no new OBV highs are found for a extended period of time (for example 8 bars), then we may want to exit from the position. The reasoning behind this is that the market may have gone stale. We could use this as a second method of exiting from positions.

CHAPTER 3

Support & Resistance Systems

Support & Resistance Systems are designed for sideways or directionless markets and they typically have the following attributes:

- They buy low and sell high in an attempt to take advantage of the sideways price movement characterizing directionless markets.
- They have a high number of winning trades, with small profits on each trade. They sell as the market goes higher and take small losses until the market finally turns down and results in a profitable trade.
- They are easier to trade emotionally.

By design, these systems miss the big move—they usually have small profits and larger losses as markets trend. The system keeps shorting a market that is in an up-trend or buying a market that is in a down-trend. Therefore, when traders use Support &Resistance systems, they use them within a group of systems that also includes trending systems and perhaps one or more volatility systems.

In this chapter, we present three Support & Resistance systems that differ in their approach but are all designed to make the most of the sideways movement of a market.

In This Chapter

Advance Decline Divergence

This is similar to the On Balance Volume: New Highs, New Lows System in that it is based on the premise that an increase in price movement will be accompanied by an increase in volume. Therefore, in order for a rally to evolve into a trend it will most likely be accompanied by an increase in volume. However, this system looks for a divergence between the price movement and the volume in order to identify tops and bottoms in price movement whereas the OBV: New Highs, New Lows system strictly looks at new highs and lows in the on balance volume to determine when an up rally is going to occur. Following is an explanation of our train of thought when developing this idea and subsequent system.

The Accumulation Distribution Indicator is an analysis technique that incorporates both price movement and volume. When the price movement for the day is positive, the daily value of the indicator is added to a running total, and when the price movement is negative, the daily value of the indicator is subtracted to this running total. The formula used by the Accumulation Distribution Indicator is as follows:

AccumDist of 1 bar ago + ((Close - Open) / (High - Low) * Volume)

As a result of the formula, when the Close of the current bar is greater than the Open, the higher the Close and the lower the Open, the more the indicator is going to advance. Likewise, when the Close is lower than the Open, the lower the Close and the higher the Open, the more the indicator will decline.

Very often, traders look at the divergence or convergence of the price movement and the Accumulation Distribution Indicator in an attempt to detect tops and bottoms in price. Yet as with many other indicators, their accurate interpretation depends on knowing the direction of the indicator. The Chaiken Oscillator is an attempt at doing this. The formula used by the Chaiken Oscillator Indicator is as follows:

Fast Exponential Average of AccumDist - Slow Exponential Average of AccumDist

The only exception to not using averages when finding tops and bottoms is when you use an average to smooth an otherwise extremely volatile or choppy indicator.

However, the averages always introduce a lag to the indicator and as a general rule are not adequate forecasting tools. When working with support and resistance (trying to pick tops and bottoms), we did not want to use averages because of the lag they introduce.

We wanted to find a reliable technique for identifying tops and bottoms. One of the most commonly-used techniques is to find convergence and divergence points between price and volume. However, volume will increase when price moves abruptly in any direction, so we need to find a way to express “negative volume,” or volume when the price declined. We could have simply used the following formula:

(Close - Open) * Volume

However, the problem with this formula is that net change in price will have a marked influence on the result; in other words, the result will change significantly based on the width of the range between the close and the open. Since our intention is to compare the price with the volume in order to find divergences, instead we used the following formula:

(Close - Open) / (High - Low) * Volume

With this new formula, the price factor will always be a number ranging from -1 to 1 regardless of the net move of the price. Because this produces a very choppy line that often crosses over and under zero, we will smooth this plot by applying a 9-bar exponential average. This formula is identical to that used by the Accumulation Distribution Indicator, except that we removed the running total.

What we are going to do then, is enter the market long when the price and the volume move show a divergence. We will buy 100 shares when the price has made a new 30-day Low and the volume move has not made a 30-day Low in the last 5 days. We will buy 200 shares when the price has made a new 30-day low and the volume move has made a new 30-day High in the last 5 days. Our thinking is that while the first scenario indicates a divergence, it is not as strong a

divergence as the second scenario. Therefore, we'll get into the market, but not expose ourselves as much as we will when the second scenario occurs.

Likewise for the short side. We will sell 100 shares when the price had made a new 30-day High and the volume move has not made a 30-day High in the last 5 days, and we will sell 200 shares when the price has made new 30-day High and the volume move has made a 30-day Low in the last 5 days.

To exit the market, we will use the same exit technique we've used for the previous systems in this volume. For our long positions, we will define an arbitrary point below the Low of our entry bar. We will subtract a 4-bar average of the range from the Low of the bar of entry, and use that as our exit point for the first bar. We will use inputs so that we can tighten this stop. Then, we will add a third of the distance between the Low of the bar and the previous exit point to the current exit point to determine the next bar's exit point. We will use this for all bars except our bar of entry.

We will do the opposite for short positions. We will add a 4-bar average of the range to the High of the bar of entry and use this as a stop for the first bar. Thereafter, we will subtract a third of the distance from the previous stop and the High of the bar from the previous stop point to determine next bar's exit price. We will use this for all bars except our bar of entry.

Figure 1 shows the AD-Divergence system applied to a chart. It also shows the AD-Div Indicator that we wrote so we could more easily visualize the behavior of our system.

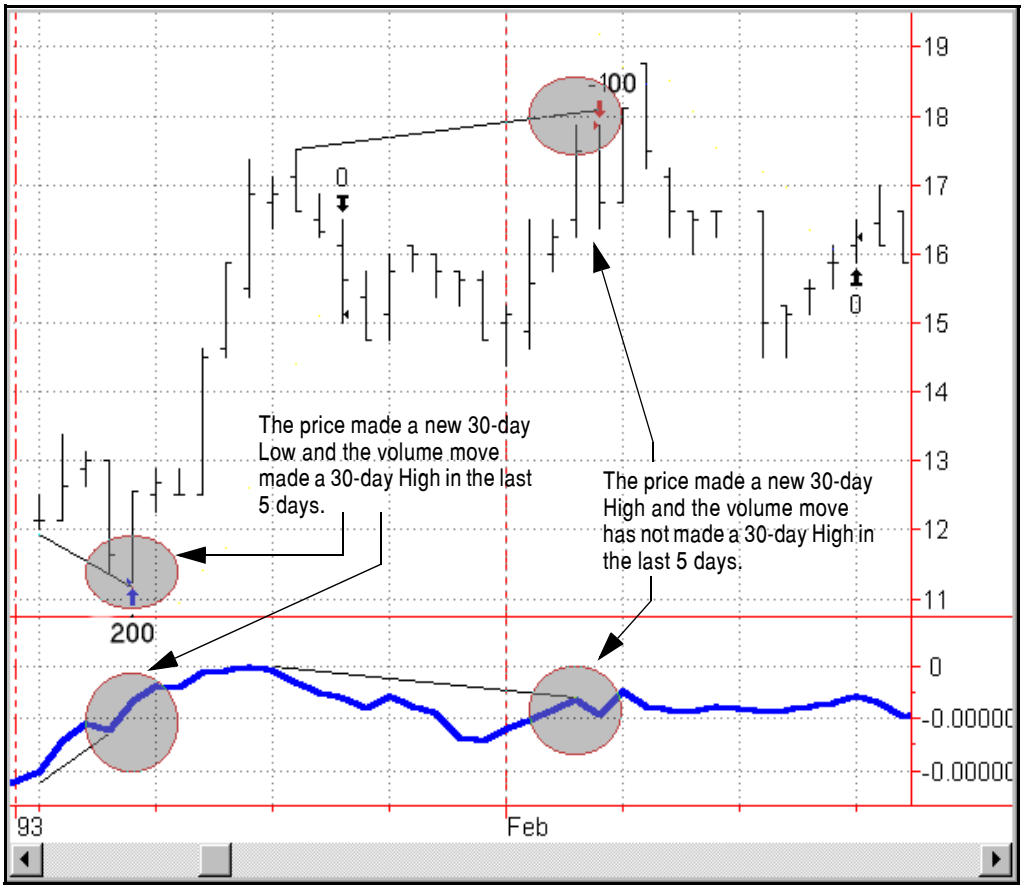


Figure 1. The AD-Divergence system applied to a chart, along with the AD-Div Indicator.

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some setup work to calculate the percent change values and the spread. The setup, entries and exits are described next:

Setup

a) Calculate the volume move on every new bar by finding an exponential average of the following formula:

(Close - Open) / (High - Low) * Volume

Long Entries

a) If the price has made a new 30-day Low and the volume move has not made a 30-day Low in the last 5 days, buy 100 shares.

b) If the price has made a new 30-day Low and the volume move made a 30-day High in the last 5 days, buy 200 shares.

Short Entries

a) If the price has made a new 30-day High and the volume move has not made a 30-day High in the last 5 days, sell 100 shares.

b) If the price has made a new 30-day High and the volume move made a 30-day Low in the last 5 days, sell 200 shares.

Exit Orders

a) When in a long position, once the bar of entry is closed, we calculate the 4-bar average of the range, divide the average by 4, and then subtract this value from the Low of the current bar. This is our long exit price for bar 2.

b) When in a long position, for all bars after bar 2, we obtain the previously calculated exit price and add to it a third of the difference between the Low and the previous stop price. We repeat this operation at the end of every bar to give us the exit price for all subsequent bars.

c) When in a short position, once the bar of entry is closed, we calculate the 4-bar average of the range and divide it by 4. Then, we add this value to the High of the current bar. This is our short exit price for bar 2.

d) When in a short position, for all bars after bar 2, we obtain the previously calculated exit price and subtract from it a third of the difference between the previous stop price and the High of the current bar. Repeating this operation at the end of every bar will give you the exit price for bar three and beyond.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Advance Decline Divergence System (STAD2: AD-Divergence)

```
Inputs: ADVol(Volume), Len(9), LookBack(30);
Vars: MP(0), StopPrice(0), VolumeMove(0), Inc(0);

{Calculates the volume move on every new bar}
VolumeMove = XAverage((Close - Open) / (High - Low) * ADVol,  Len);

{Checks for divergence between price movement and volume move and places buy & sell orders}
If BarNumber > 1 then Begin
        If Low < Lowest(Low , LookBack)[1] and MRO( VolumeMove < Lowest(VolumeMove ,
                LookBack )[1],5,1) = -1 thenBuy 100 shares next bar at Market;
        If Low < Lowest(Low, LookBack)[1] and VolumeMove > Highest(VolumeMove, LookBack)[1]
                then Buy 200 shares next bar at Market;
        If High > Highest(High, LooKBack)[1] AND MRO(VolumeMove > Highest( VolumeMove,
                LookBack)[1],5,1) = -1 then Sell 100 shares next bar at Market;
        If High > Highest(High, LookBack)[1] and VolumeMove < Lowest(VolumeMove, LookBack)[1]
                then Sell 200 shares next bar at Market;
End;

{Determines market position and calculates stop prices for first bar in position}
MP = MarketPosition;
If MP = 1 and MP[1] <> 1 then Begin
        StopPrice = Low - Average(Range,4);
        Inc = (Low - StopPrice )/3;
End;
If MP = -1 and MP[1] <> -1 then Begin
        StopPrice = High + Average(Range,4);
        Inc =  (StopPrice - High)/3;
End;

{Places exit orders and calculates stop prices for bars after first bar in position}
If MP = 1 then Begin
        Exitlong ("ExitLong") next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice )/3;
End;
If MP = -1 then Begin
        Exitshort ("Exitshort") next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default Value	Description
ADVol	Volume	Volume to be used. If working with futures in intraday bars, Ticks can be used instead of Volume.
Len	9	Number of bars to be used when calculating the exponential moving average.
LookBack	30	Number of bars to look bar when establishing new highs and new lows.

In addition to these inputs, we define the following variables:

```
Vars: MP(0), StopPrice(0), VolumeMove(0), Inc(0);
```

Then we calculate the volume move and assign it to the variable VolumeMove:

```
VolumeMove = XAverage((Close - Open) / (High - Low) * ADVol,  Len);
```

Long Entries

If there is a new 30-bar low and in the last 5 bars there has been no new lows in the VolumeMove then buy 100 shares next bar at the open.We use the MRO function to determine whether there have been new lows:

```
If Low < Lowest( Low , LookBack )[1] and MRO(VolumeMove < Lowest(VolumeMove,
        LookBack )[1],5,1) = -1 then Buy 100 shares next bar at Market;
```

If there is a new 30-bar low and in the last 5 bars there has been a new high in the VolumeMove then buy 200 shares next bar at the open:

```
If Low < Lowest( Low, LookBack )[1] and VolumeMove > Highest( VolumeMove,
        LookBack)[1] then Buy 200 shares next bar at Market;
```

Short Entries

If there is a new High and in the last 5 bars there has been no new highs in VolumeMove then sell 100 shares next bar at the open:

```
If High > Highest( High, LooKBack )[1] AND MRO(VolumeMove > Highest( VolumeMove,
        LookBack)[1],5,1) = -1 then Sell 100 shares next bar at Market;
```

If there is a new High and in the last 5 bars there has been a new low in VolumeMove then sell 200 shares next bar at the open:

```
If High > Highest( High, LookBack )[1] and VolumeMove < Lowest( VolumeMove, LookBack)[1]
        then Sell 200 shares next bar at market;
```

Exit Orders

At the first bar of a long position, the system will define StopPrice as the low of the current bar minus a 4-bar average of the range:

```
If MP = 1 and MP[1] <> 1 then Begin
        LastOrderBar = -999;
        StopPrice = Low - Average(Range,4);
End;
```

At the first bar of a short position, the system will define StopPrice as the high of the current bar plus a 4-bar average of the range:

```
If MP = -1 and MP[1] <> -1 then Begin
        StopPrice = High + Average(Range,4);
        LastOrderBar = -999;
End;
```

If in a long position, then we will place an order to exit at the stop price or anything lower. Once this is done, the stop price for the next bar is calculated as the current stop price plus a third of the difference between the low and the current stop price.

```
If MP = 1 then Begin
        Exitlong ("paralong") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

If in a short position, we will place an order to exit at the stop price or anything higher. Once this is done, the stop price for the next bar is calculated as the current stop price minus a third of the difference between the current stop price and the high.

```
If MP = -1 then Begin
        Exitshort ("parashort") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We did not include margin because we designed this system for stocks, and we specified 1 as the default number of contracts to trade per order. The system is written such that it will buy (or sell) 100 or 200 shares depending on the criteria that is met. This overrides the setting in this tab.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. In this system, this is determined by the EasyLanguage instructions, as the number of shares is hard-coded.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 2 shows a sample system report for the Advance Decline Divergence system. In this system report, we see characteristics that are typical of trending systems. The percent of wining trades is under 50% yet the average win/loss ratio is close to 2.

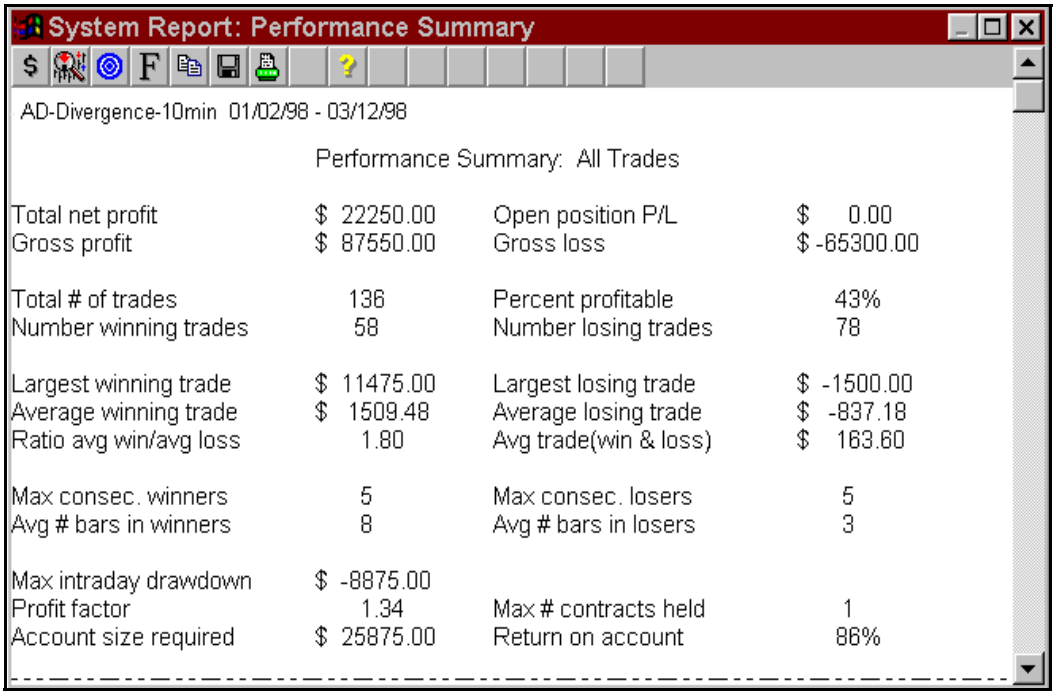


Figure 2. Sample system report for Advance Decline Divergence system.

This system traded quite frequently, producing 136 trades in three months of 10-minute bars. In order to test a system appropriately, and have the test be statistically significant, we want to have *at least* 25 instances of each order of our system. Yet the number of orders is not the only factor that we want to consider when reviewing system results; another crucial factor, and one which is overlooked surprisingly often, is that the data on which the system is tested must be representative of all the different modes that the market can adopt. The current system report shows the results for three months, which is a period of time that might not represent all the different market modes.

Suggestions for Improvement

If you are trading futures, you will not have the volume available to you until the following day. Instead, you will have available the number of ticks; however, this number can be deceiving because one tick represents a transaction, not how many contracts were moved by that transaction. To avoid being deceived by an incorrect value, when trading futures, you can simply remove the Volume section from the formula and use:

VolumeMove = (Close - Open) / (High - Low)

The results will look different (between -1 and 1) but you will not be comparing price movement to number of transactions.

Stochastics

The Stochastics Indicator is a commonly-used overbought/oversold analysis technique used to determine when the market is reaching a top or bottom. However, it occurs frequently that this oscillator reaches the over bought area and the underlying symbol continues its rally, riding its momentum for several points more, which keeps the oscillator in this over bought area for some time before the trend finishes.

Many times, we have seen the Stochastic Indicator reach this overbought area so we place a short order only to watch the market continue to move up for some time. To avoid this, we will lower the overbought band and establish a long position when the K component of the

Stochastics Indicator (SlowK) reaches this band and then wait for the market to show signs of reversing before establishing our short position.

One SlowK reaches the overbought area, we will start calculating a trailing stop that will not only exit from the long position, but will reverse and go short. Because we are buying when our indicator is signaling an overbought market, we will buy 1 contract (or 200 shares if trading stocks), and when our trailing stop is hit, we will reverse and go short with 2 contracts (or 400 shares).

Our trailing stop will work as follows: at the bar of entry, we will find the 4-bar average of the range and we will subtract this value from the low, and this will be our reversal price. On every consecutive bar we will calculate the distance from the low to the reversal price of the previous bar and divide it into 3, we will add the result of this operation to the previous bar's reversal price. As soon as this reverse price is hit, we will exit our long position and reverse.

To cover our long position, we will place a trailing stop one tick under the lowest low of the last five bars.

The opposite will be true for the short side of our system. We will up the oversold band and establish a short position when the SlowK reaches this band and then wait for the market to show signs of reversing before establishing our long position.

One the SlowK reaches the oversold area, we will start calculating a trailing stop that will not only exit from the short position, but will reverse and go long. Because we are selling when our indicator is signaling an oversold market, we will sell 1 contract (or 200 shares if trading stocks), and when our trailing stop is hit we will reverse and go long with 2 contracts (or 400 shares).

Our trailing stop will work as follows: at the bar of entry, we will find the 4-bar average of the range and we will add this value to the high, and this will be our reversal price. On every consecutive bar, we will calculate the distance from the reversal price of the previous bar to the high and divide it into 3, then we will subtract the result of this operation from the previous bar's reversal price. As soon as this reverse price is hit, we will exit our short position and reverse.

To cover our short position, we will place a trailing stop 1 tick under the lowest low of the last 5 bars.

Figure 3 shows this system applied to a daily chart. The Xs are simply a visual tool we used to identify the position of the trailing stop. Notice that at Point A, since the SlowK had been greater the overbought line for the last 3 bars, we entered the market with a long position and placed the trailing stop. Then, on the very next bar, the trailing stop was hit so we reversed (and doubled) our position.

Then, we exited our short position at Point B, when the price penetrated the highest high of the last 5 bars. At Point C, we went long again when the SlowK had been greater than the overbought line for the last 3 bars. We reversed (and doubled) our position when the price penetrated the reversal price (Point D).

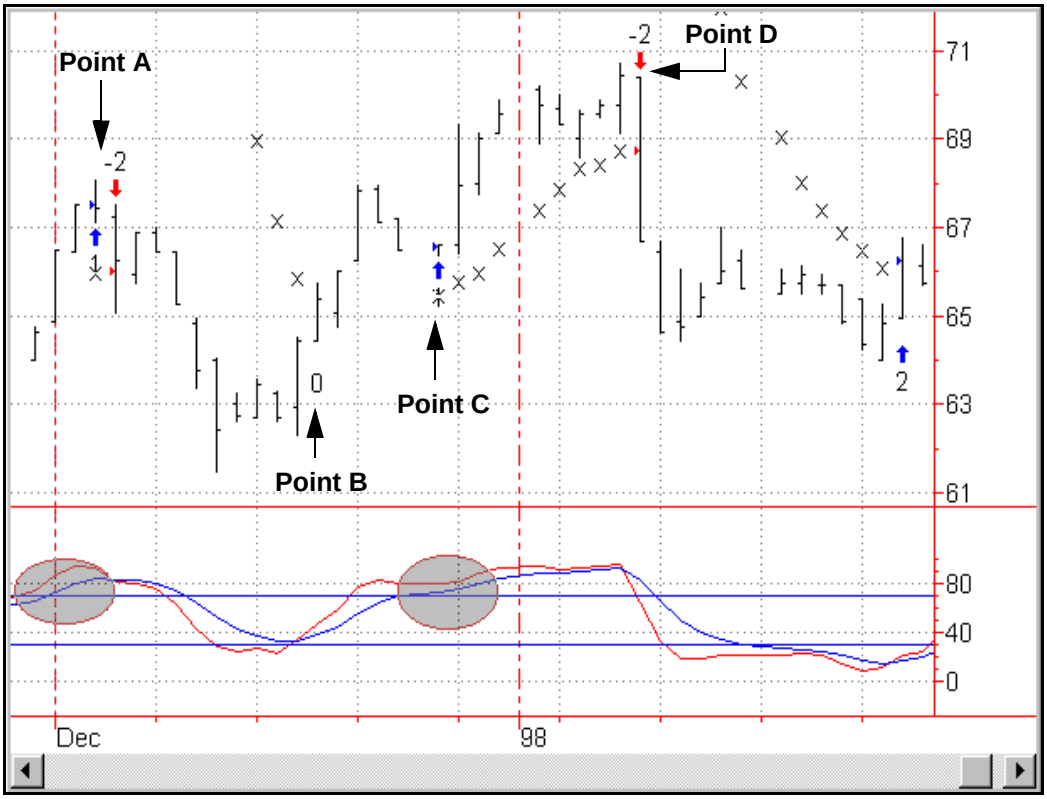


Figure 3. The Stochastics system applied to a chart.

Defining your Trading Rules

In this system we defined both long and short entries as well as exit orders. The long and short entries will reverse your position, whereas the exits will close out your existing position and exit you from the market. There was essentially no setup work to do. The entry and exits are described next:

Long Entries

a) If the SlowK has been greater than the overbought level for 3 bars and you are currently out of any positions, then buy 200 shares.

b) If the system is in a short position as a result of the SlowK being under the oversold band for 3 or more bars and the trailing stop for the short position is hit, then establish the long position with 400 shares.

Short Entries

a) If the SlowK has been lower than the oversold level for 3 bars and you are currently out of any positions, then sell 200 shares.

b) If the system is in a long position as a result of the SlowK being greater than the overbought band for 3 or more bars and the trailing stop for the long position is hit, then establish a short position with 400 shares.

Exits

a) Exit out of any long positions initiated by the reversal order when the market falls and penetrates a 5-bar Low channel

b) Exit out of the short positions initiated by the reversal orders when the market rallies and breaks a 5-bar High channel

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Stochastics System (STAD2: Stochastics)

```
Input: Length(14), OBLevel(75), OSLevel(25);
Vars: MP(0), Mode(0), ReversePrice(0);
MP = MarketPosition;

{Determines whether the SlowK value has been less than the overbought level for the last 3 bars
and places buy order}
If MRO(SlowK( Length ) < OBLevel,3,1) = -1  then Begin
        If MarketPosition <> -1 then Begin
                Buy 200 shares next bar at market;
                Mode = 1;
        End;
End;

{Calculates reversal price and places sell stop order based on new reversal price}
If MP = 1 and MP[1] <> 1 then ReversePrice = Low - Average(Range,4);

If MP = 1 and Mode = 1 then Begin
        Sell ("ReverseLong") 400 shares next bar at ReversePrice stop;
        ReversePrice = ReversePrice + (Low - ReversePrice)/3;
End;

{Exit order for short positions}
ExitShort from entry ("ReverseLong") at Highest(High, 5) stop;

If MRO(SlowK( Length ) > OSLevel,3,1) = -1 then Begin
        If MarketPosition <> 1 then Begin
                Sell 200 shares next bar at Market;
                Mode = -1;
        End;
End;

{Calculates reversal price and places buy stop order based on new reversal price}
If MP = -1 and MP[1] <> -1 then ReversePrice = High + Average(Range,4);

If MP = -1 and Mode = -1 then Begin
        Buy ("ReverseShort") 400 shares next bar at ReversePrice stop;
        ReversePrice = ReversePrice - (ReversePrice - High)/3;
End;

{Exit order for long positions}
ExitLong from entry ("ReverseShort") at Lowest(Low, 5) stop;
```

Inputs

For this system, we will use the following inputs:

Input	Default Value	Description
Length	14	Number of bars used when calculating stochastics
OBLevel	75	Overbought level for stochastics
OSLevel	25	Oversold level for stochastics

In addition to these inputs, we define the following variables:

```
Vars: MP(0), Mode(0), ReversePrice(0);
```

Long Entries

IMPORTANT: The MRO function returns the number of the bar on which the conditions most recently occurred. When it returns -1, it means the condition did not occur.

Therefore, we are checking to make sure the SlowK has NOT been under the OBLevel during the last 3 bars, in which case it has been equal to or over.

If the SlowK has been greater than the overbought level for 3 consecutive bars, and the system is not holding a short position, then we will buy 200 shares next bar at the open. Additionally, we will use the variable called Mode as a switch, and set it to 1.

```
If MRO(SlowK( Length ) < OBLevel,3,1) = -1  then Begin
        If MarketPosition <> -1 then Begin
                Buy 200 shares next bar at market;
                Mode = 1;
        End;
End;
```

If the system is in the first bar of short position, then we calculate the reverse price as the High plus the 4-bar average of the range.

```
If MP = -1 and MP[1] <> -1 then ReversePrice = High + Average(Range,4);
```

If the system is in a short position, that was initiated by the SlowK being in the over sold area for 3 consecutive bars (Mode = -1) then we will look to buy 400 shares next bar at the reverse price defined above. And we will set the reverse price for the next bar as the reverse price minus a third of the reverse price minus the High.

```
If MP = -1 and Mode = -1 then Begin
        Buy ("ReverseShort") 400 shares next bar at ReversePrice stop;
        ReversePrice = ReversePrice - (ReversePrice-high)/3;
End;
```

Short Entries

If the SlowK is under the over sold band for 3 consecutive bars, and the system is not in a long position, then it will short 200 shares of the stock. It will also set the variable Mode to -1.

```
If MRO(SlowK( Length) > OSLevel,3,1) = -1 then Begin
        If MarketPosition <> 1 then Begin
                Sell 200 shares next bar at Market;
                Mode = -1;
        End;
End;
```

If the system is in the first bar of long position, then we calculate the reversal price as the low minus the 4-bar average of the range.

```
If MP = 1 and MP[1] <> 1 thenReversePrice = Low - Average(Range,4);
```

If the system is in a long position initiated by the SlowK being in the over bought area for 3 consecutive bars (Mode = 1) then we will look to sell 400 shares next bar at the reverse price defined above. Additionally, we will set the reverse price for the next bar as the reverse price plus a third of the difference between the low and the reverse price.

```
If MP = 1 and Mode = 1 then Begin
        Sell ("ReverseLong") 400 shares next bar at StopPrice stop;
        ReversePrice = ReversePrice + (Low-ReversePrice)/3;
End;
```

Exit Orders

Exit out of any long positions initiated by the reversal order when the market falls and penetrates a 5-bar Low channel:

```
ExitShort from entry ("ReverseLong") at Highest(High, 5) stop;
```

Exit out of the short positions initiated by the reversal orders when the market rallies and breaks a 5-bar High channel:

```
ExitLong from entry ("ReverseShort") at Lowest(Low, 5) stop;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We did not include margin because we designed this system for stocks, and we specified 1 as the default number of contracts to trade per order. The system is written such that it will buy (or sell) 200 or 400 shares depending on the criteria that is met. This overrides the setting in this tab.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. In this system, this is determined by the EasyLanguage instructions, because the number of shares is hard-coded.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 4 shows a sample system report for the Stochastics System. This system report shows typical numbers for a Support and Resistance system. Notice that the percentage of profitable trades is near 50% and the win/loss average ratio is above 1 yet not spectacular by any means.

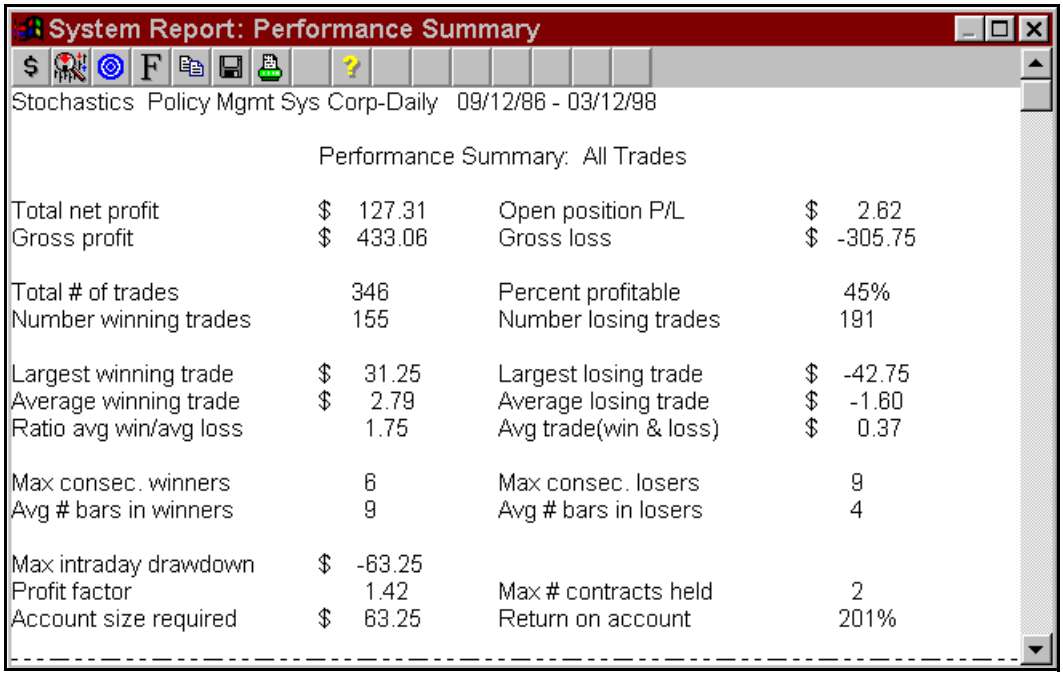


Figure 4. Sample system report for Stochastics system

One statistic that jumps out in this report is the size of the largest losing trade. The system had a loss of 42-3/4 points in one trade while the average losing trade was 1-5/8 points. This would lead us to believe that our protective stops failed to detect this trade and may fail again in the future. These extremely large losing trades are probably the most important trades we will find when back testing any system as they can tell us how and when our system is failing to protect its capital.

Suggestions for Improvement

If the system has a low percentage profitable trades, we can try removing the initial buy when the SlowK is greater than the overbought band for 3 bars, and simply enter the short position when the reversal order is hit.

Likewise, we may want to try removing the initial sell order when the SlowK is lower than the oversold band for 3 bars, and simply enter the long position when the reversal order is hit.

CCI Bollinger Bands - Overbought and Oversold

Bollinger Bands are a very popular technical analysis tool often used to determine when the market is trading outside of the 'normal' trading range. Bollinger Bands are lines plotted +2 and -2 standard deviations above and below a moving average. In data that is distributed normally, 99% of prices is enclosed within these 2 standards deviations of the average price. Therefore, we can safely assume that prices that trade outside of the Bollinger Bands are significant.

In trending markets, prices trading outside of the Bollinger Bands usually signify that the market has a great deal of momentum and is moving strongly with the trend. In choppy markets, Bollinger Bands can indicate that the market is oversold or overbought and is bound to change direction.

Because of this dual interpretation, we will use a secondary condition to confirm the relationship of the price to the Bollinger Bands. To do so, we will use the CCI Indicator. The CCI Indicator is an oscillator that indicates overbought and oversold conditions in the market when the indicator is over 100 or under -100, respectively. The CCI is designed so that about 75% of the trading activity falls in the neutral territory. We will look for this oscillator to turn

up while it is in the oversold area and for it to turn down when it is in the overbought area and consider these to be significant conditions for this system.

Combining these two analysis techniques, CCI and Bollinger Bands, will give us our setup and entry conditions. When the price activity has recently crossed under the lower Bollinger Band and the CCI is in the oversold territory and turns up, we will buy. If the price has recently traded over the Bollinger Band and the CCI is in the overought territory and turns down, we will sell.

Figure 5 shows this system applied to a chart. We also applied the Bollinger Bands and CCI Indicators to the chart, so we could more easily visualize the system's behavior.

When you apply an indicator that performs the same calculation as your system to a chart, make sure the inputs are set the same for both. Also, make sure the maximum number of bars the study will reference (MaxBarsBack) is also set the same.

If you the inputs and/or MaxBarsBack setting are different, the results will be different and you may think your system is not calculating correctly, when it actually is.

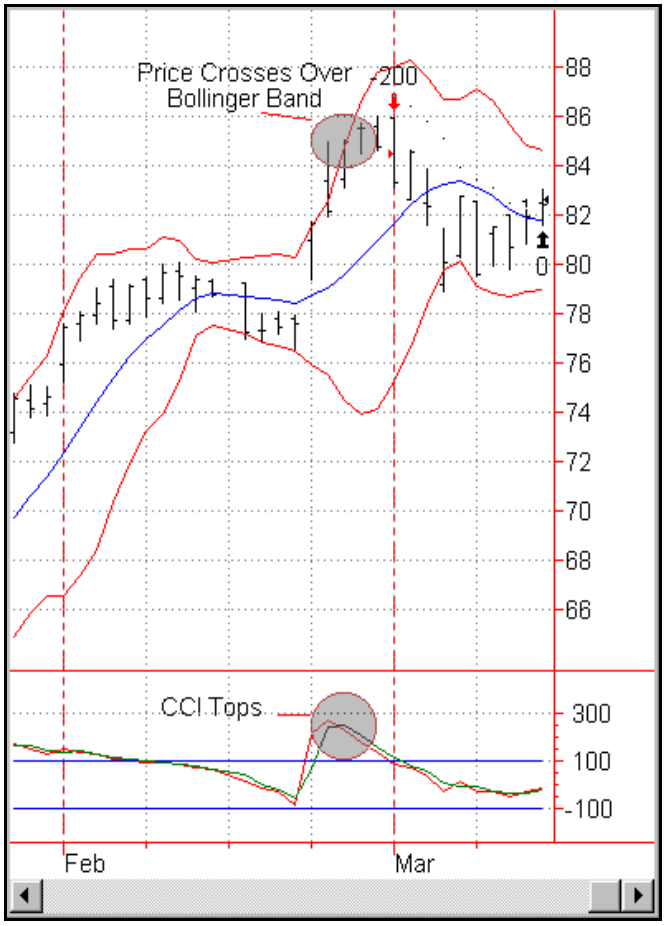


Figure 5. The CCI Bollinger - Overbought Oversold system applied to a chart

Now we need a method of exiting the market. We will work with a variation of a trailing stop exit technique used in the previous volume of STAD Club. For our long positions, we will define an arbitrary point below the Low of our entry bar. In this system, we will subtract the average of the range of the last 4 bars from the Low of the bar of entry and use that as our exit point for the first bar. From then on, we will add a third of the distance between the Low of the bar and the previous exit point to determine our next day's exit point.

We will do the opposite for our short positions. We will add a 4-bar average of the range to the High of the bar of entry and use this as a stop for the first bar. Thereafter, we will subtract a third of the distance from the previous stop and the High of the bar from the previous stop point to determine next bar's exit price.

This will work for any bar except for our bar of entry. To cover our entry bar, when in a long position, we will risk the lowest Low of the last 6 bars, and when in a short position, we will risk the highest High of the last 6 bars.

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. We also performed some setup work, which involved calculating the Bollinger Bands and determining where the prices of the current bar are in relation to the bands. The long and short entries will reverse your position, whereas the exits will close out your existing position and exit you from the market. The setup, entry and exits are described next:

Setup

a) We will calculate the position of the Close in regard to the Bollinger Bands by using the following formula:

(Close - Bottom Band) / (Top Band - Bottom Band)

b) If Close crosses over or under 0.5, then we will not produce any trades.

c) If the High of the current bar crosses over the top band, we will look for short positions.

d) If the Low of the current bar crosses under the bottom band, we will look for long positions.

Long Entries

a) If the CCI is declining and turns up while in the oversold band, we will find the highest High of the last 3 bars. This will be our entry price

b) During the next 5 bars, we will look for the price to rally and buy at our entry price.

Short Entries

a) If the CCI is rising and turns down while in the overbought band, we will find the lowest Low of the last 3 bars. This will be our entry price.

b) During the next 5 bars, we will look for the price to decline and establish a short position at our entry price.

Exits

a) Once the bar of entry of a long position is closed, calculate the 4-bar average of the range and divide it by 4. We will then subtract this value from the low. This will be our long exit price for bar 2.

b) From the second bar, we obtain the exit price calculated in the last point and add to it a third of the difference between the Low and the previous stop price. Repeating this operation at the end of every bar will give us the exit price for bar 3 and beyond.

c) At the bar of entry of a short position, we will place an exit order for short positions at the highest High of the last 8 bars.

d) Once the bar of entry of a short position is closed, calculate the 4-bar average of the range and divide it by 4. We then add this value to the High. This will be our short exit price for bar 2.

e) From the second bar, we obtain the exit price calculated in the last point and add to it a third of the difference between the Low and the previous stop price. Repeating this operation at the end of every bar will give us the exit price for bar 3 and beyond.

f) Place a money management protective stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts are being traded.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: CCI Bollinger Bands - Overbought and Oversold System (STAD2: CCi Bollinger)

```
Inputs: BBLen(9), BBStDv(2),CCILen(14), CCIAvgLn(2);
Vars: Position(0), TopBand(0), BotBand(0),BBStatus(0), CCIAvg(0), CITurnsUp(False),
        CCITurnsDown(False), StopPrice(0), CP(0), LongCount(0), ShortCount(0), LEP(0), SEP(0);

{Calculate Bollinger Bands and CCI}
TopBand = BollingerBand(Close, BBLen, BBStDv);
BotBand = BollingerBand(Close, BBLen, -BBStDv);
CCIAvg = XAverage(CCI(CCILen), CCIAvgLn);

{Determines location of prices in relation to bands}
If TopBand <> BotBand then Position = (Close - BotBand) / (TopBand - BotBand);
        If CurrentBar > 1 then Begin
        If Position crosses over .5 then BBStatus = 0;
        If Position crosses under .5 then BBStatus = 0;
        If Low crosses under BotBand then BBStatus = -1;
        If High crosses over TopBand then BBStatus = 1;
End;

{Determines location of CCI}
CCITurnsUp = SwingLowBar(1, CCIAvg, 1, 3) = 1 AND CCIAvg[1] < -100;
CCITurnsDown = SwingHighbar(1, CCIAvg, 2, 3) = 2 AND CCIAvg[2] > 100;

{Calculates long entry price}
If BBStatus = -1 and CCITurnsUp  then Begin
        LongCount = CurrentBar;
        LEP = Highest(High,3);
End;

{Places long entry order, keeping it active for 5 bars}
If CurrentBar < LongCount + 5 and MarketPosition <> 1 then Buy next bar at LEP stop;

{Calculates short entry price}
If BBStatus = 1 and CCITurnsDown then Begin
        ShortCount = CurrentBar;
        SEP = Lowest(low,3);
End;

{Places short entry order, keeping it active for 5 bars}
If CurrentBar < ShortCount + 5 and MarketPosition <> -1 then Sell next bar at SEP stop;

{Changes value of LongCount variable when in a long or short position, to avoid placing another
buy or sell order}
If MarketPosition = 1 then LongCount = -999;
If MarketPosition = -1 then LongCount = -999;

{Calculates stop price and places exit orders}
CP = MarketPosition;
if CP = 1 and CP[1] <> 1 then StopPrice = Low - Average(Range,4)*2/3;
If CP = -1 and CP[1] <> -1 then StopPrice = High + Average(Range,4)*2/3;
If CP = 1 then Begin
        Value99 = Text_New(Date,Time,StopPrice,".");
        Exitlong ("ExitLong") next bar at StopPrice stop;
        StopPrice = StopPrice + ( .3 *(Low - StopPrice));
```

```
End;
If CP = -1 then Begin
        Value99 = Text_New(Date,Time,StopPrice,".");
        Exitshort ("ExitShort") next bar at stopprice stop;
        StopPrice = StopPrice - (.3 * (StopPrice - High));
End;
```

Inputs

For this system, we will use the following inputs:

Input	Default Value	Description
BBLen	9	Number of bars used to calculate Bollinger bands
BBStDv	2	Number of standard deviations used to calculate Bollinger Bands
CCILen	14	Number of bars used to calculate CCI
CCIAvgLn	2	Number of bars used in the average of the CCI

Additionally we will declare the following variables:

```
Vars: Position(0), TopBand(0), BotBand(0), BBStatus(0), CCIAvg(0), CCITurnsUp(false),
        CCITurnsDown(false); Vars: StopPrice(0), CP(0), LongCount(0), ShortCount(0),
        LEP(0), SEP(0);
```

Setup

We will first define the TopBand and BotBand variables as our Bollinger Bands, and the CCIAvg variable as the exponential average of the CCI.

```
TopBand = BollingerBand(Close, BBLen, BBStDv);
BotBand = BollingerBand(Close, BBLen, -BBStDv);
CCIAvg = XAverage(CCI(CCILen), CCIAvgLn);
```

We will first define the variable Position as the position where the close is with respect to the two bands (checking for division by zero first):

```
If TopBand <> BotBand then Position = (Close - BotBand) / (TopBand - BotBand);
```

Then we look for the Position variable to cross over or under the middle distance of the two Bollinger Bands (Position crosses over 0) and for the Low to cross under the bottom band or the High to cross over the top band:

```
If CurrentBar > 1 then Begin
        If Position crosses over .5 then BBStatus = 0;
        If Position crosses under .5 then BBStatus = 0;
        If Low crosses under BotBand then BBStatus = -1;
        If High crosses over TopBand then BBStatus = 1;
End;
```

Finally, we will look for the CCI average to turn up while it is under -100, or for the CCI average to turn down while it is over 100. The result of these true/false expressions are stored in the variables CCITrunsUp and CCITurnsDown, respectively.

```
CCITurnsUp = SwingLowBar(1, CCIAvg, 1, 3) = 1 AND CCIAvg[1] < -100;
CCITurnsDown = SwingHighbar(1, CCIAvg, 2, 3) = 2 AND CCIAvg[2] > 100;
```

Long Entries

When the Low has crossed under the bottom band (BBStatus = -1) and the CCI has turned up, we will store the current bar's bar number in the LongCount variable and we will define our long entry price as the highest High of the last three bars. We store this long entry price in the variable LEP.

```
If BBStatus = -1 and CCITurnsUp  then Begin
        LongCount = CurrentBar;
        LEP = Highest(high,3);
End;
```

If the High crossed over the top Bollinger Band and the CCI turned up in the last 5 bars, then we will buy next bar at our long entry price.

```
If Currentbar < LongCount + 5 and MarketPosition <> 1 then Buy next bar at LEP stop;
```

Short Entries

When the high has crossed over the top band (BBStatus = 1) and the CCI has turned down then we will hold the current bar's bar number in the ShortCount variable and we will define our short entry price as the lowest Low of the last 3 bars.

```
If BBStatus = 1 and CCITurnsDown  then Begin
        ShortCount = CurrentBar;
        SEP = Lowest(Low,3);
End;
```

If the High crossed under the bottom Bollinger Band and the CCI turned down in the last 5 bars, then we will sell next bar at our short entry price.

```
If CurrentBar < ShortCount + 5 and MarketPosition <> -1 then Sell next bar at SEP stop;
```

Exit Orders

At the first bar of a long position, the system will define StopPrice as the Low of the current bar minus a 4-bar average of the range.

```
If MP = 1 and MP[1] <> 1 then Begin
        LastOrderBar = -999;
        StopPrice = Low - Average(Range,4);
End;
```

At the first bar of a short position, the system will define StopPrice as the High of the current bar plus a 4-bar average of the range.

```
If MP = -1 and MP[1] <> -1 then Begin
        StopPrice = High + Average(Range,4);
        LastOrderBar = -999;
End;
```

If in a long position, we will place an order to exit at the stop price or anything lower. Once this is done, the stop price for the next bar is calculated as the current stop price plus a third of the difference between the Low and the current stop price.

```
If MP = 1 then Begin
        Exitlong ("ExitLong") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low- StopPrice)/3;
End;
```

If in a short position, we will place an order to exit at the stop price or anything higher. Once this is done, the stop price for the next bar is calculated as the current stop price minus a third of the difference between the current stop price and the High.

```
If MP = -1 then Begin
        Exitshort ("ExitShort") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We did not include margin because we designed this system for stocks, and we specified 1as the default number of contracts/shares to trade per order.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. In this system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

The CCI-Bollinger system analyzes support and resistance levels and has unusual results for a system of this nature. Figure 6 shows a sample System Report. Notice that the percentage of wining trades is typical (if not low at 46%) yet the ratio of average winning versus losing trades is very high at 2.5. This high ratio could be attributed to an effective exit technique. It produced around 160 trades in 12 years for this stock, which would amount to about one trade every month, so the trading system is not over-trading. Also, the largest winning trade and the largest losing trade are very low compared to the net gain, and closer to the average winning and losing

trade, respectively. This would indicate that the system is performing consistently throughout the entire sample data used for this test.

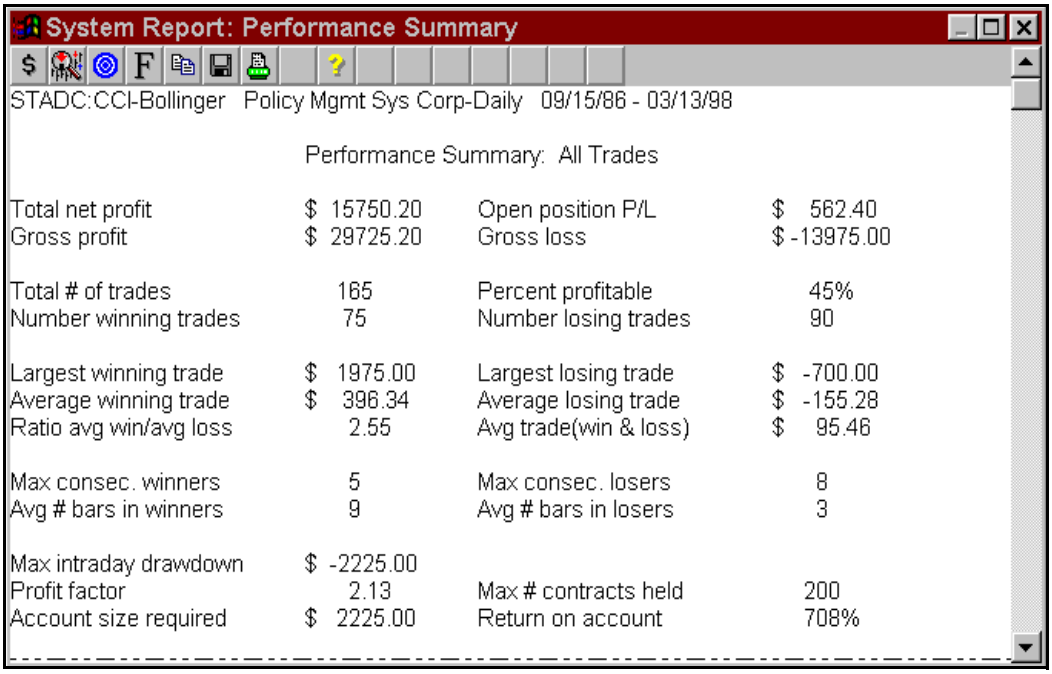


Figure 6. Sample system report for the CCI Bollinger - Overbought Oversold system

Suggestions for Improvement

We may find that this system is too sensitive (trades too often) or is missing too many moves (rarely trades). In this case, we may want to try adjusting the number of standard deviations we use to calculate Bollinger Bands. Using a higher number will make the system trade less often since we are moving the bands away from the price.

CHAPTER 4

Volatility Breakout Systems

Volatile markets are characterized by sharp jumps in price, and volatility breakout systems are designed to take advantage of this type of change in volatility. Volatility breakout systems generally have the following characteristics:

- Substantial amount of time out of the market.
- High percentage of winning trades, but with a small profit per trade.
- Don't take advantage of big moves.
- Exciting to trade because trades are quick and short-term.
- Based solely on price movement.

When designing this type of system, the key is to effectively anticipate and take advantage of a significant change in volatility and then exit the position before a loss of profit. In this chapter, we present three volatility breakout systems, all of which are designed to capture significant change in volatility and limit losses during directionless and/or trending phases. We also present a somewhat different section on determining short-term market outlook

In This Chapter

Percent Change Channel

One of the most important concepts in trading is volatility—specifically, how markets react to extreme moves. The age-old principle that states that for every action there is a reaction also applies to trading. When there is a sudden extreme move in one direction, the market will tend to make a correction. It is very hard to catch the initial move and act upon it, but it is not so hard to catch and act upon the correction that usually follows these sudden moves. This is the basis for this system.

We will look for the percent change of the last 5 closes to be greater than 102% (2 percent increase in the last 5 bars) and then place an order to buy at the Low of the bar. Likewise, when the percent change of the Close is lower than 98% (2 percent decrease), we will place an order to sell at the High of the bar. These orders will remain active for 3 bars, at which point they will be cancelled if they haven't been filled.

Because we want to take a quick profit, we will exit at the first open that gives us a profit: an open greater than our entry price in long positions and an open lower than our entry price for short positions. Also, to exit quickly from any bad positions, we will exit out of our long positions at the lowest Low of the last five bars and we will exit out of our short positions at the highest High of the last five bars.

Figure 1 shows the system applied to a chart.

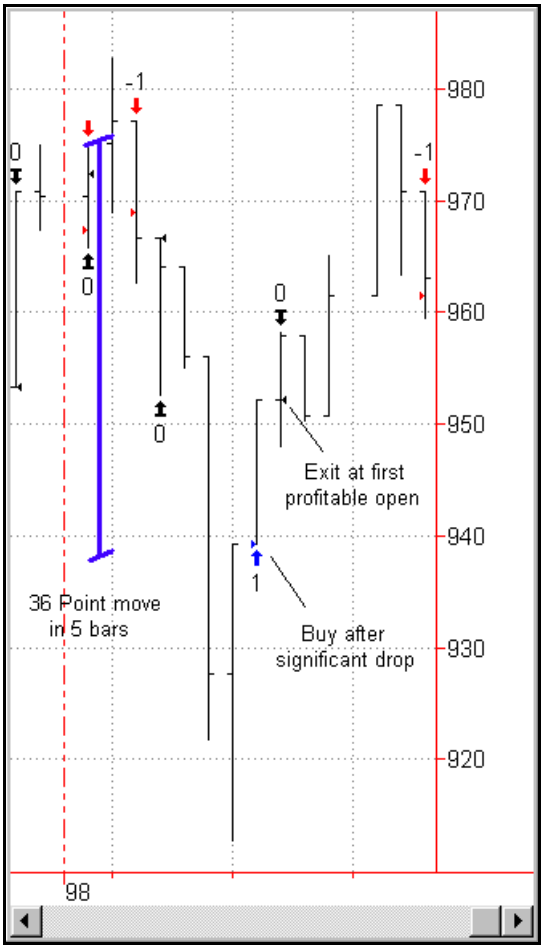


Figure 1. The Percent Change Channel System applied to a chart

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. We also did some setup work which included calculating the percent change. How we calculated the setup, entries and exits is described next.

Setup

a) We will calculate the percent change as the Close of the current bar divided by the Close of 5 bars ago and multiply the result by 100.

Long Entries

a) If the price has declined more than 2%, then we will place an order to buy at the High of the bar that declined 2% or anything higher. This order will remain active for 3 bars.

Short Entries

a) If the price has increased more than 2%, then we will place an order to sell short at the Low of the bar that declined 2% or anything lower. This order will remain active for 3 bars.

Exits

a) We will exit from both long and short positions at the first profitable open.

b) We will exit from any long positions at the lowest Low of the last 5 bars.

c) We will exit form any short positions at the highest High of the last 5 bars.

d) We will place a money management stop for both long and short positions.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Percent Change Channel System (STAD2: % Change Chnl)

```
Input: Pchng(5), OBought(102), OSold(98), Bands(3);
Vars: PcntChange(0), HighBand(0), LowBand(0), OS(FALSE), OB(FALSE);

{Calculates the percent change}
PcntChange = Close / Close[PChng] * 100;

{Compares the percent change to the oversold level and calculates the long entry price}
If PcntChange < OSold then Begin
        OS = TRUE;
        HighBand = Highest( High , Bands );
End;

{Compares the percent change to the overbought level and calculates the short entry price}
If PcntChange > OBought then Begin
        OB = TRUE;
        LowBand = Lowest( Low, Bands);
End;
```

```
{Places short and long entry orders}
If OS and MarketPosition <> 1 Then Buy next bar at Highband stop;
If OB and MarketPosition <> -1 then Sell next bar at lowband stop;

{Places exit orders and resets OS and OB variables}
If MarketPosition = 1 then If open next bar > Entryprice then exitlong next bar at open;
if MarketPosition = -1 thenIf open next bar < Entryprice thenexitshort next bar at open;

If MRO( OS = False, 2, 1 ) = -1 then OS = False;
If MRO( OB = False, 2, 1 ) = -1 then OB = False;

Exitlong at Lowest(Low,5) stop;
Exitshort at Highest(High,5) stop;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default Value	Description
PChng	5	Number of bars to use when calculating percent change
OBought	102	Percent over which we will consider the market had a significant move
OSold	98	Percent under which we will consider the market had a significant move
Bands	3	Number of bars used to determine entry price in both long and short price

We will also declare the following variables:

```
Vars: PcntChange(0), HighBand(0), LowBand(0), OS(FALSE), OB(FALSE);
```

Setup

We calculate the percent change using the following EasyLanguage instructions:

```
PcntChange = Close / Close[PChng] * 100;
```

Long Entries

If the percent change is under the OSold value (set by default to 98), then we will set the variable OS to TRUE and we will define our entry price as the highest High of the last 3 bars:

```
If PcntChange < OSold then Begin
        OS = TRUE;
        HighBand = Highest( High , Bands );
End;
```

If the OS variable is True and we are not in a long position, then we will buy on the next bar at our entry price (HighBand) or anything higher:

```
If OS and MarketPosition <> 1 Then Buy next bar at Highband stop;
```

If the variable OS has been True for the last 2 bars, then we will set it to False. Thus cancelling the order placed above.

```
If MRO( OS = False, 2, 1 ) = -1 then OS = False;
```

Short Entries

If the percent change is over the OBought input (set as default as 102) then we will set the variable OB to TRUE and we will define our entry price as the lowest low of the last 3 bars.

```
If PcntChange > OBought then Begin
        OB = TRUE;
        LowBand = Lowest( Low, Bands);
End;
```

If the OB variable is true and we are not in a short position then we will sell short next bar at our entry price (LowBand) or anything lower.

```
If OB and MarketPosition <> -1 then sell next bar at LowBand stop;
```

If the variable OB has been true for the last 2 bars, then we will set it to False. Thus cancelling the order placed above. This is an effective way of keeping the order active for only 3 bars.

```
If MRO( OB = False, 2, 1 ) = -1 then OB = false;
```

Exit Orders

```
If MarketPosition = 1 then if Open next bar > entryprice then Exitlong next bar at open;
If MarketPosition = -1 then if Open next bar < entryprice thenEexitshort next bar at open;
Exitlong at Lowest(Low,5) stop;
Exitshort at Highest(High,5) stop;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We specified 1 as the default number of contracts to trade per order.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. In this system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 2 shows a sample System Report for the Percent Change Channel System. As you can see, this system shows results typical of a volatility expansion system, with a very high percent of profitable trades at 68% and a low win/loss ratio at .77. Notice also how the largest winning trades represents nearly 11% of the total net profit, and most importantly, how the maximum losing trade is just one point away from the average winning trade. This leads us to believe that the system has a very stable and effective method of detecting bad trades and cutting the losses early.

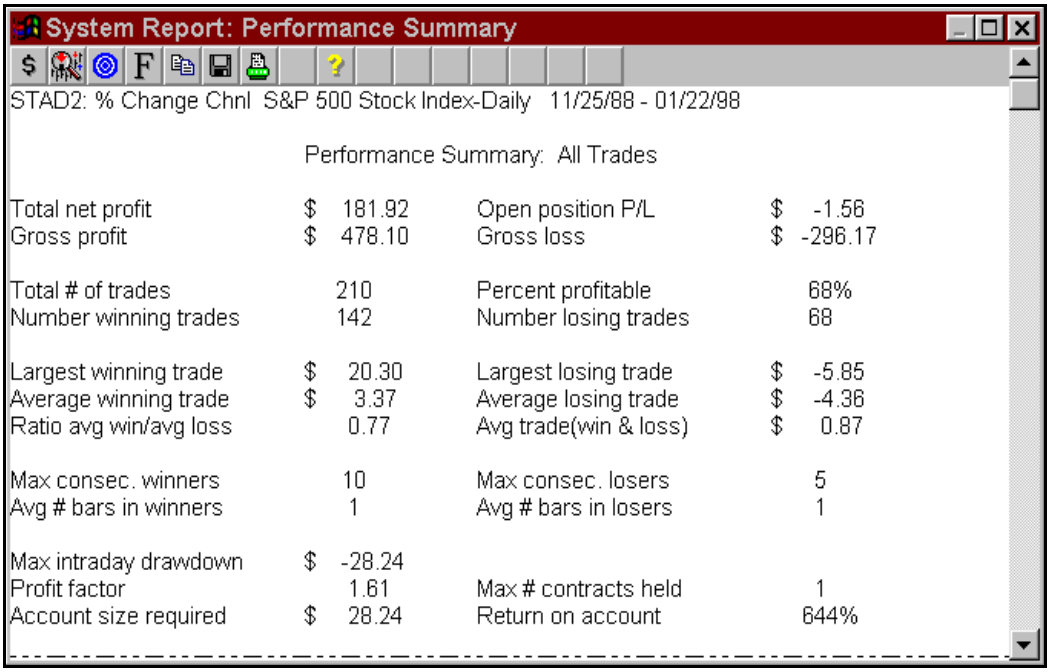


Figure 2. A sample System Report for the Percent Change Channel System

Suggestions for Improvement

The more prone the symbol is to volatility and violent moves, the greater number we will want to use for the OBBand and the smaller the number for OSBand. Moving these numbers away from 100 will reduce the number of trades but will probably increase the profitability and percent of winning trades.

Bull Beater

This system is based on an idea provided by Jack Cahn.

A strategy commonly-used to trade a volatile market is to identify quiet or sideways periods in an attempt to capitalize on up or down market moves. As discussed in Chapter 3, when describing the CCI-Bollinger System, Bollinger Bands is a very popular technical analysis tool often used to determine when the market is trading outside of 'normal' trading ranges. Bollinger Bands are 2 lines, each plotted 2 standard deviations above or below a moving average. In data that is normally distributed, 99% of the prices are enclosed within these 2 standards deviations, so we could safely assume that prices that trade outside of the Bollinger Bands are significant.

Since we can use Bollinger Bands to determine when prices are moving outside of normal ranges, we can also use them to determine when prices are trading within the normal range (moving sideways). As we look at Bollinger Bands in a chart, we will notice that the bands move away from the price when the price becomes volatile (or gaps) and that they move closer to the price as it stabilizes.

We will do a few things then, to determine when we are in a sideways period. First, we will look for the Bollinger Bands to get closer to the price bars. We will write an indicator called STAD2: BullBeater to easily see when this is occurring. The indicator will subtract the bottom Bollinger Band from the top Bollinger Band and display the result as a histogram. We will also look for low volatility and define low volatility as being when 1.5 times the average range for the last 4 bars is greater than the difference between the two Bollinger Bands.

In addition, we can determine when the market is oversold or overbought by looking at the RSI level when the market falls into this sideways period. If the RSI is greater than 55, we will consider the market as prone to fall and place an order to buy at a channel break out. If the RSI is less than 45, we will consider the market as prone to rise and place an order to sell short at a channel break under. Also, night activity (or second session price movement) is very different, so we will only trade a symbol's day session.

Now that we have defined our entries, we will define the exits. For our long positions, we will define an arbitrary point below the Low of our entry bar. In this system, we will subtract a 4-bar average of the range from the Low of the bar of entry, and use that as our exit point for the first bar. We will use inputs so that we can tighten this stop. Then, we will add a third of the distance between the Low of the bar and the previous exit point to the current exit point to determine the next bar's exit point. We will use this for all bars except our bar of entry. To cover our entry bar, we will risk a predetermined amount by using a stop loss (money management stop).

We will do the opposite for short positions. We will add a 4-bar average of the range to the High of the bar of entry and use this as a stop for the first bar. Thereafter, we will subtract a third of the distance from the previous stop and the High of the bar from the previous stop point to determine next bar's exit price. We will use this for all bars except our bar of entry. For our bar of entry, we will place a stop order at the highest High of the last 8 bars.

Figure 3 shows the BullBeater system applied to a chart along with the Bollinger Bands Indicator and the RSI Indicator.

Figure 3. The BullBeater System applied to a chart

Defining your Trading Rules

In this system, we defined both long entries and short entries as well as exit orders. Again, the long and short entries reverse your position, whereas the exits will close out your existing position and exit you from the market. There was really no setup involved for this system other than calculating the Bollinger Bands and RSI values as well as defining the trading time. The setup, entries and exits are described next.

Setup

a) Calculate the Bollinger Bands.

b) Calculate the RSI values.

c) Define our trading time as after 9:30am and before 4:30pm.

Long Entries

a) If 1.5 times the 4-bar average of the range is greater than the difference between the two Bollinger Bands and the RSI is in the oversold area, then we will place an order to buy at the highest High of the last 5 bars. The order will remain active for 5 bars.

Short Entries

a) If 1.5 times the 4-bar average of the range is greater than the difference between the two Bollinger Bands and the RSI is in the overbought area, then we will place an order to sell at the lowest Low of the last 5 bars. The order will remain active for 5 bars.

Exits

a) Once the bar of entry of a long position is closed, calculate the 4-bar average of the range and divide it by 4, then subtract this value from the Low. This will be our long exit price for bar 2.

b) From the second bar, we obtain the exit price calculated in the last point and add to it a third of the difference between the Low and the previous stop price. Repeating this operation at the end of every bar will give us the exit price for all subsequent bars.

c) At the bar of entry of a short position, we will place an short exit order at the highest High of the last 8 bars

d) Once the bar of entry of a short position is closed, calculate the 4-bar average of the range and divide it by 4, then add this value to the High. This will be our short exit price for bar 2.

e) From the second bar, we obtain the exit price calculated in the last point and add to it a third of the difference between the Low and the previous stop price. Repeating this operation at the end of every bar will give us the exit price for all subsequent bars.

f) We will also place a money management protective stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts we are trading.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: BullBeater System (STAD2: BullBeater)

```
Input: Length(9), SDev(2), OSold(55), OBought(45), StartTme(930), EndTme(1630);
Vars: TBB(0), BBB(0), MP(0), StopPrice(0), TradeTime(False), OS(False), OB(False);

{Calculates the Bollinger Bands, RSI and determines if it is currently the day session}
TBB = BollingerBand(C,Length, SDev);
BBB = BollingerBand(C, Length, -SDev);
OB = RSI(C,4) > OSold;
OS = RSI(C,4) < OBought;
TradeTime =  Time > StartTme and time < EndTme;

{Determines if volatility is low}
If (TBB-BBB) < 1.5*Average(Range,4) then Value1 = BarNumber;

{Places buy and sell orders}
If BarNumber < Value1 +  5 and TradeTime then Begin
        If OS then Buy next bar at Highest(High,5) stop;
        If OB then Sell next bar at Lowest(Low,5) stop;
End;

{Calculates stop price and places exit orders}
MP = MarketPosition;
If MP = -1 and MP[1] <> -1 then StopPrice = High + Average(range,4);
if MP = 1 and MP[1] <> 1 then StopPrice = Low - Average(range,4);

If MP = 1 then Begin
        Exitlong ("paralong") next bar at StopPrice stop;
        StopPrice = stopprice + (Low - StopPrice)/3;
End;
If MP = -1 then Begin
        Exitshort ("parashort") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice- High)/3;
End;
```

Inputs

Following is the list of all the inputs we used in this system:

Input	Default Value	Description
Length	9	Number of bars used to calculate Bollinger Bands
SDev	2	Number of standard deviations used for Bollinger Bands
OSold	55	Value at which RSI is considered at an over sold level
OBought	45	Value at which RSI is considered at an over bought level
StartTme	930	Time at which the system will start looking at trading rules

Input	Default Value	Description
EndTme	1630	Time at which the system will stop looking at trading rules

Setup

We will calculate the Bollinger Bands and store them into variables.

```
TBB = BollingerBand(C,Length, SDev);
BBB = BollingerBand(C, Length, -SDev);
```

We will then calculate the RSI and determine if it is overbought or oversold as well as see if the current time is within our defined trading time. We will store the result of these expressions in the OB, OS and TradeTime variables:

```
OB = RSI(C,4) > OSold; OS = RSI(C,4) < OBought;
TradeTime =  Time > StartTme and time < EndTme;
```

If we find that the difference between the Bollinger bands is less than 1.5 times the 4-bar average of the range, then we will store the bar number in the variable Value1:

```
If (TBB-BBB) < 1.5*Average(Range,4) then Value1 = BarNumber;
```

Long Entries

If less than 5 bars have closed since the Bollinger Bands narrowed (indicating low price movement), the time of the current bar is within our trading times, and the RSI is in the oversold area, we will buy at the highest High of the last 5 bars:

```
If BarNumber < Value1 + 5 and TradeTime then Begin
        If OS then Buy next bar at Highest(High,5) stop;
        If OB then Sell next bar at Lowest(Low,5) stop;
End;
```

Short Entries

If less than 5 bars have closed since the Bollinger Bands narrowed (indicating low price movement), the time of the current bar is within our trading times, and the RSI is in the overbought area. we will sell short at the lowest Low of the last 5 bars:

```
If BarNumber < value1 + 5 and TradeTime then Begin
        If OS then Buy next bar at Highest(High,5) stop;
        If OB then Sell next bar at Lowest(Low,5) stop;
End;
```

Exit Orders

At the first bar of a long position, the system will define StopPrice as the Low of the current bar minus a 4-bar average of the range.

```
If MP = -1 and MP[1] <> -1 then StopPrice = High + Average(Range,4);
```

At the first bar of a short position, the system will define StopPrice as the High of the current bar plus a 4-bar average of the range.

```
If MP = 1 and MP[1] <> 1 then StopPrice = Low - Average(Range,4);
```

If in a long position, we will place an order to exit at the stop price or anything lower. Once this is done, the stop price for the next bar is calculated as the current stop price plus a third of the difference between the Low and the current stop price:

```
If MP = 1 then Begin
        Exitlong ("paralong") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

If in a short position, we will place an order to exit at the stop price or anything higher. Once this is done, the stop price for the next bar is calculated as the current stop price minus a third of the difference between the current stop price and the High:

```
If MP = -1 then Begin
        Exitshort ("parashort") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

When we apply this system to a chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for commission and slippage. We specified 1 as the default number of contracts to trade per order.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. In this system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the **Do not allow multiple entries in same direction** option. If the system is in a long position and market conditions generate another long entry order, the order is ignored. This is also the case when we're in a short position and market conditions generate another short entry order.

Testing & Improving

Figure 4 shows a sample System Report for the BullBeater System. As is the case with most volatility expansion systems, this System Report shows a high percentage of winning trades of short duration, with a low ratio between average winning and loosing trades. The percentage of winning trades is near 60% and the win/loss ratio is under 1.

In spite of the high percentage of winning trades, it is very disturbing that all of the net profit is coming from the largest winning trade. This can be corrected in one of two ways, improving the winners or limiting the losing trades. This system seems to have very good accuracy (60% of winning trades), so the obvious area that we should work on is reducing our losses. It seems necessary to study the losing trades to try to minimize both their occurrence and the average loss they incur.

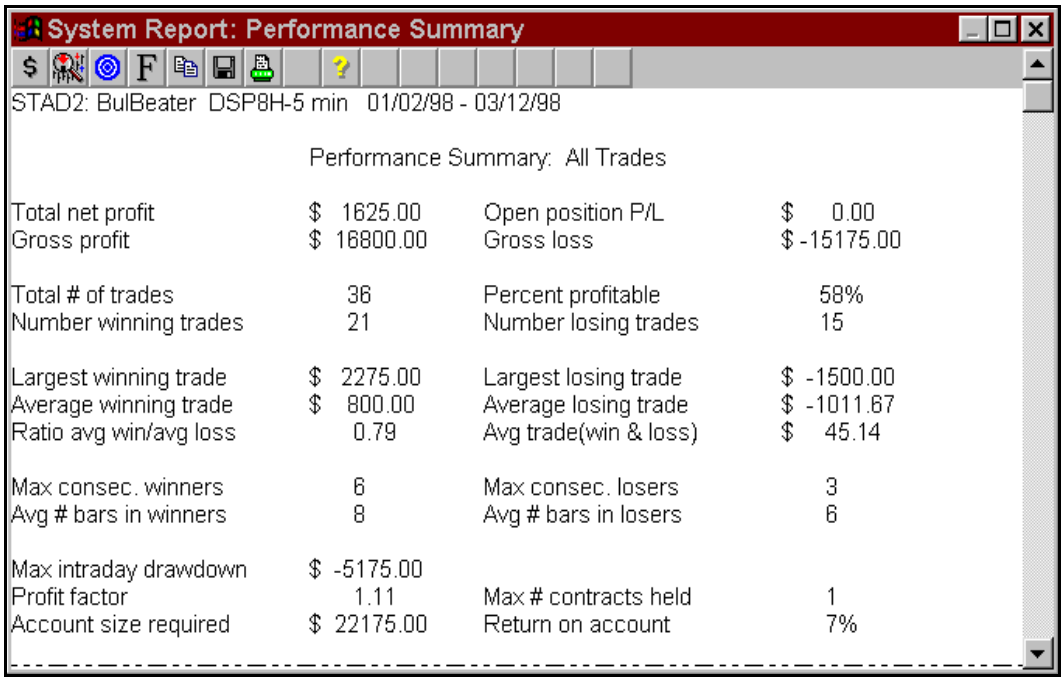


Figure 4. A sample System Report for the BullBeater System

Suggestions for Improvement

When the market falls into a congestion area (sideways mode), it will eventually break out, either on the up side or on the down side. If the market is overbought, it will probably decline, but chances are that this period is just a correction to the up trend (or as it is called, the market breathing before continuing the trend) so we would miss this move if it did happen. We may want to try placing an buy order farther away from the market when the RSI is in the overbought area and a short order when the RSI is in the oversold area.

Opening Gap

Opening Gap, like Percent Change Channel, is another system that looks at the volatility of a symbol and how markets react to extreme moves; however it is based on the opening gap and designed for intraday charts.

In designing this system, we decided to look at the opening gap at the market's open. In other words, when the market opens above the previous day's high or below the previous day's Low. This indicates a sudden, extreme move.

Our assumption will be that if the market is in a long-term up trend and it opens for the day under the previous day's Low, then it will try to correct this with a rally. Likewise, if the market is in a long-term down trend and it opens for the day over the previous day's High, it will try to correct this with a decline.

Therefore, we will look at a fast and slow exponential moving average of daily bars to determine the direction of the market in daily bars. When the fast daily average is over the slow daily average, then the market is in a bullish trend. This means we will need to create a chart with two data streams. The symbol is plotted as Data1,as usual, with an intraday data compression, and the same symbol is plotted again in the same chart as Data2, with a daily data compression.

If the market is in a bullish trend and it opens under the Low of the previous day, we will buy at the first price available. Additionally, it may happen that the market continues to move down because of the momentum caused by the down gap, so we will place a trailing buy throughout the day: we will buy whenever the price breaks the highest High of the last bar.

If the market is in a bearish trend and it opens over the High of the previous day, we will sell at the first price available. Additionally, it may happen that the market continues to move up because of the momentum caused by the down gap, so we will place a trailing sell throughout the day: we will sell whenever the price breaks the lowest Low of the last bar.

Now we need a method of exiting the market. We will work with a variation of the trailing stop exit technique we used in most of the other systems in this volume. For our long positions, we will define an arbitrary point below the Low of our entry bar. We will subtract the average of the range of the last 4 bars from the Low of the bar of entry and use that as our exit point for the first bar. From there on, we will add a third of the distance between the Low of the bar and the previous exit point to determine the exit point for all subsequent bars.

We will do the opposite for short positions. We will add a 4-bar average of the range to the High of the bar of entry and use this as a stop for the first bar. Thereafter, we will subtract a third of the distance from the previous stop to the High of the bar from the previous stop point to determine the exit point for all subsequent bars.

This will work for any bar except for our bar of entry. To cover our entry bar, we will place a stop loss 2 points away from the entry price.

Figure 5 shows the Opening Gap system applied to a chart. This chart only shows one data stream; the other is hidden.

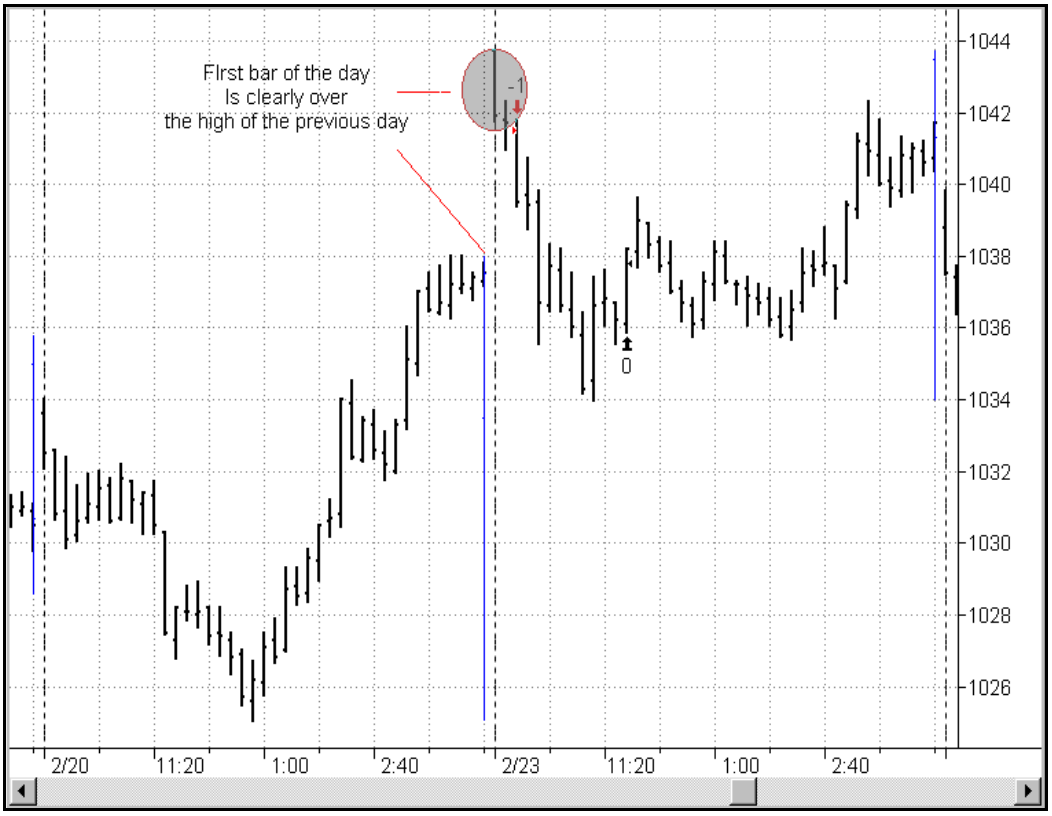


Figure 5. The Opening Gap System applied to a chart

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. The long and short entries will reverse your position, whereas the exits will close out your existing position and exit you from the market. We did some simple setup work which involved calculating the exponential averages. The setup, entries and exits are described next:

Setup

a) We will determine there is a bull market whenever the 3-day exponential average is greater than the 6-day exponential average.

b) We will determine there is a bear market whenever the 3-day exponential average is lower than the 6-day exponential average.

Long Entries

a) If we have determined the market is bullish, we will buy if the first bar of the day closes below the Low of the previous day.

b) If the Close of the first bar of the day is below the previous days Low, we will buy whenever the High of the previous intraday bar is broken.

Short Entries

a) If we have determined the market is bearish, we will sell if the first bar of the day closes above the High of the previous day.

b) If the Close of the first ar of the day is above the previous day's High, we will sell whenever the low of the previous intraday bar is broken.

Exits

a) Once the bar of entry of a long position is closed, calculate the 4-bar average of the range and divide it by 4, then subtract this value from the Low. This will be our long exit price for bar 2.

b) From the second bar, we obtain the exit price calculated in the last point and add to it a third of the difference between the Low and the previous stop price. Repeating this operation at the end of every bar will give us the exit price for all subsequent bars.

d) Once the bar of entry of a short position is closed, calculate the 4-bar average of the range and divide it by 4, then add this value to the High. This will be our short exit price for bar 2.

e) From the second bar, we obtain the exit price calculated in the last point and add to it a third of the difference between the Low and the previous stop price. Repeating this operation at the end of every bar will give use the exit price for all subsequent bars.

f) We will place a money management protective stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts are being traded.

Designing & Formatting

This section presents the EasyLanguage instructions and formatting for the system, with the EasyLanguage instructions broken down and explained line by line.

EasyLanguage Instructions: Opening Gap System (STAD2: Opening Gap)

```
Vars: ShortDayAvg(0), LongDayAvg(0), StopPrice(0), BullTrend(False), BearTrend(True),
        SellOnSwingHigh(False), BuyOnSwingLow(False), MP(0);

{Calculates the exponential averages and compares them to determine bearish or bullish trend}
ShortDayAvg = XAverage(Close, 3) of Data2;
LongDayAvg = XAverage(Close, 6) of Data2;
BullTrend = ShortDayAvg >= LongDayAvg;
BearTrend = ShortDayAvg < LongDayAvg;

{On first bar of the day, places buy and sell orders when additional criteria is met}
If Date <> Date[1] then Begin
        SellOnSwingHigh = False;
        BuyOnSwingLow = False;
        If BullTrend then Begin
                If Close < Low of Data2 then Begin
                        Buy this bar at the close;
                        BuyOnSwingLow = True;
                End;
                If Close > High of Data2 then SellOnSwingHigh = True;
        End;
        If BearTrend then Begin
                If Close > High of Data2 then Begin
                        Sell this bar at the close;
                        SellOnSwingHigh = True;
                End;
                If Close < Low of Data2 then BuyOnSwingLow = True;
        End;
End;

{Places buy and sell orders when additional criteria is met}
If Date = Date[1] Then Begin
        If SellOnSwingHigh then Sell next bar at Low[1] stop;
        If BuyOnSwingLow then Buy next bar at High[1] stop;
        If MarketPosition = -1 then SellOnSwingHigh = False;
        If MarketPosition = 1 then BuyOnSwingLow = False;
End;

{Calculates stop prices and places exit orders}
MP = MarketPosition;
If MP = 1 and MP[1] <> 1 then StopPrice = Low - Average(Range,4);
If MP = -1 and MP[1] <> -1 then StopPrice = High + Average(Range,4);

If MP = 1 then begin
        Exitlong ("ExitLong") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;

If MP = -1 then Begin
        Exitshort ("ExitShort") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

Inputs

This system uses no inputs. However, we will declare the following variables:

```
Vars: ShortDayAvg(0), LongDayAvg(0), StopPrice(0), BullTrend(false), BearTrend(true),
      SellOnSwingHigh(false), BuyOnSwingLow(false), MP(0);
```

Setup

We will define the variables ShortDayAvg and LongDayAvg as the 3- and 6-day exponential averages of the Close of Data2. Data2 should be daily bars of the symbol defined as Data1.

```
ShortDayAvg = XAverage(Close, 3) of Data2;
LongDayAvg = XAverage(Close, 6) of Data2;
```

Additionally, we will define a bull trend whenever the short average is greater or equal to the long average, and a bear trend whenever the short average is lower than the long average:

```
BullTrend = ShortDayAvg >= LongDayAvg;
BearTrend = ShortDayAvg < LongDayAvg;
```

Long Entries

When a new day starts, we will set the variable BuyOnSwingLow to FALSE. Also, if we detect a bull trend, and the Close is lower than the daily Close, we will buy at the Close of the first intraday bar. We will also set the variable BuyOnSwingLow to TRUE.

However, if we detect a bear trend, and the Close is lower than the Low of Data2, we will set the variable BuyOnSwingHigh to TRUE.

```
If Date <> Date[1] then Begin
        SellOnSwingHigh = False;
        BuyOnSwingLow = False;
        If BullTrend then Begin
                If Close < Low of Data2 then Begin
                        Buy this bar at the close;
                        BuyOnSwingLow = True;
                End;
                If Close > High of Data2 then SellOnSwingHigh = True;
        End;
        If BearTrend then Begin
                If Close > High of Data2 then Begin
                        Sell this bar at the close;
                        SellOnSwingHigh = True;
                End;
                If Close < Low of Data2 then BuyOnSwingLow = True;
        End;
End;
```

After the first bar of the day is closed (Date = Date[1]) then we will look to buy at the High of the previous bar if the condition BuyOnSwingLow is true. BuyOnSwingLow will be true whenever the Close of the first bar of the day is lower than the Low of the previous day. Also, because we only want one trade during the day, once we are in a long position, we will set this variable to false:

```
If Date = Date[1] Then Begin
        If SellOnSwingHigh thenSell next bar at Low[1] stop;
        If BuyOnSwingLow then Buy next bar at high[1] stop;
        If MarketPosition = -1 then SellOnSwingHigh = False;
        If MarketPosition = 1 then BuyOnSwingLow = False;
End;
```

Short Entries

When a new day starts, we will set the variable SellOnSwingHigh to FALSE. Then, if we detect a bear trend, and the Close is higher than the daily Close, we will sell at the Close of the first intraday bar. We will also set the variable SellOnSwingHigh to TRUE.

If we detect a bear trend, and the Close is higher than the High of Data2, we will set the variable SellOnSwingLow to TRUE.

```
If Date <> Date[1] then Begin
        SellOnSwingHigh = False;
        BuyOnSwingLow = False;
        If BullTrend then Begin
                If Close < Low of Data2 then Begin
                        Buy this bar at the close;
                        BuyOnSwingLow = True;
                End;
                If Close > High of Data2 then SellOnSwingHigh = True;
        End;
        If BearTrend then Begin
                If Close > High of Data2 then Begin
                        Sell this bar at the close;
                        SellOnSwingHigh = True;
                End;
                If Close < Low of Data2 then BuyOnSwingLow = True;
        End;
End;
```

After the first bar of the day is closed (Date = Date[1]) then we will look to sell at the Low of the previous bar if the condition SellOnSwingHigh is true. SellOnSwingHigh will be true whenever the Close of the first bar of the day is higher than the High of the previous day. Also, because we only want one trade during the day, once we are in a short position, we will set the SellOnSwingHigh variable to False.

```
If Date = Date[1] Then Begin
        If SellOnSwingHigh then Sell next bar at Low[1] stop;
        If BuyOnSwingLow then Buy next bar at high[1] stop;
        If MarketPosition = -1 then SellOnSwingHigh = False;
        If MarketPosition = 1 then BuyOnSwingLow = False;
End;
```

Exit Orders

At the first bar of a long position, the system will define StopPrice as the low of the current bar minus a 4-bar average of the range:

```
If MP = 1 and MP[1] <> 1 then StopPrice = Low - Average(Range,4);
```

At the first bar of a short position, the system will define StopPrice as the high of the current bar plus a 4-bar average of the range:

```
If MP = -1 and MP[1] <> -1 thenStopPrice = High + Average(Range,4);
```

If in a long position, then we will place an order to exit at the stop price or anything lower. Once this is done, the stop price for the next bar is calculated as the current stop price plus a third of the difference between the Low and the current stop price.

```
If MP = 1 then Begin
        Exitlong ("ExitLong") next bar at StopPrice stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

If in a short position, we will place an order to exit at the stop price or anything higher. Once this is done, the stop price for the next bar is calculated as the current stop price minus a third of the difference between the current stop price and the High.

```
If MP = -1 then Begin
        Exitshort ("ExitShort") next bar at StopPrice stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

General System Format

Again, this system was designed to be applied to a multi-data chart, with intraday bars as Data1 and daily bars as Data2. When we apply this system to our multi-data chart, we use the options in the **Format** dialog box to format it as follows:

a) In the **Costs** tab, we entered the appropriate amounts for the default number of contracts to trade, commission and slippage.

Note: *Remember that commissions are calculated on a per contract/share basis. When you are trading stocks, you would enter the average commission you are charged divided by the number of shares the system is buying and selling. Unless specified within the system, this is determined by the* ***Default Contracts*** *option on this tab.*

b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option can hold the dollar amount per position or dollar amount per contract/share you want to risk before exiting out of the position.

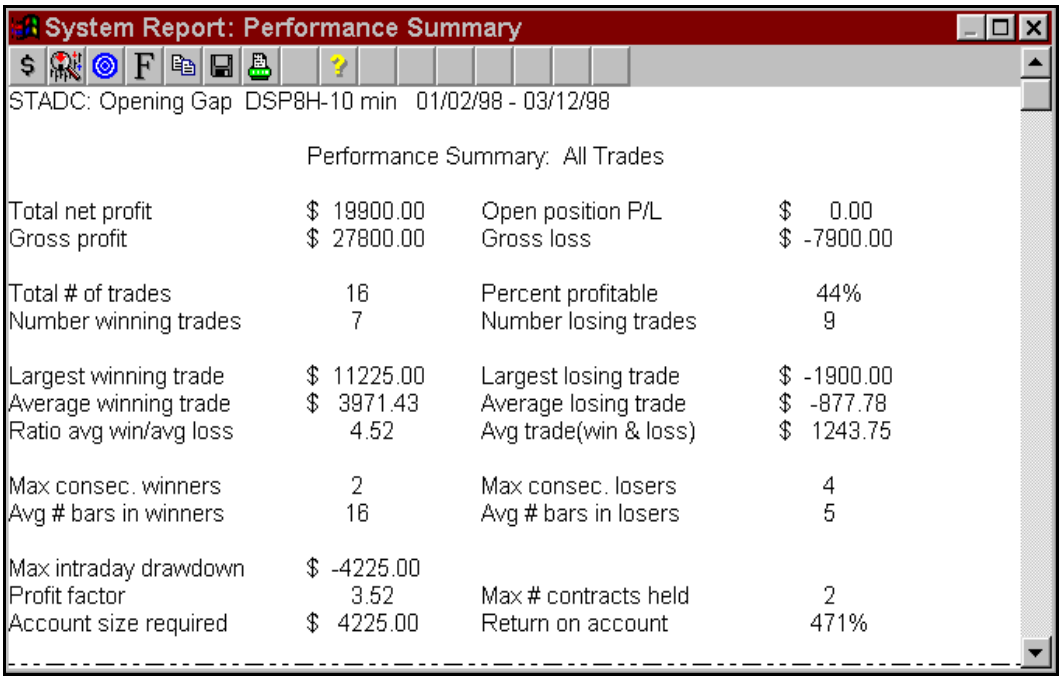
Note: *When you are trading stocks and you choose the stop to be tracked on a per share (contract) basis, you will type in the number of points you are willing to lose before you are exited out. When you are trading futures or any instrument that has a different dollar-point value, you would type the maximum number of dollars you are willing to risk per contract traded.*

c) In the **Properties** tab, we selected the option **Allow Multiple Entries in the same direction by same and different entry orders**. This is because this system is designed to scale into a position.

d) Also in the **Properties** tab, we typed the number 4 in the option **Maximum number of entries per position**. This limits the number of times the system can enter the market in the same direction to 4. For example, if we type 4 in this option and the system is buying blocks of 100 shares, the system to hold a maximum of 400 shares at any given time.

Testing & Improving

Figure 6 shows a sample System Report for the Opening Gap system. This system has very positive numbers in this report—while the average winning versus losing trades are near 50%, it has a very high win/loss ratio at 4.5. This System Report shows the results of the last 3 months and only has 16 trades, which means this system trades infrequently (approximately once a week). This is an important factor to consider; we will have to be very patient to trade this system.



System Report: Performance Summary

STADC: Opening Gap DSP8H-10 min 01/02/98 - 03/12/98

Performance Summary: All Trades

Total net profit	$ 19900.00	Open position P/L	$ 0.00
Gross profit	$ 27800.00	Gross loss	$ -7900.00
Total # of trades	16	Percent profitable	44%
Number winning trades	7	Number losing trades	9
Largest winning trade	$ 11225.00	Largest losing trade	$ -1900.00
Average winning trade	$ 3971.43	Average losing trade	$ -877.78
Ratio avg win/avg loss	4.52	Avg trade(win & loss)	$ 1243.75
Max consec. winners	2	Max consec. losers	4
Avg # bars in winners	16	Avg # bars in losers	5
Max intraday drawdown	$ -4225.00		
Profit factor	3.52	Max # contracts held	2
Account size required	$ 4225.00	Return on account	471%

Figure 6. A sample System Report for the Opening Gap System.

Again, this system trades infrequently by design as it is looking for gap-up and gap-down days, which are not every day events. The number of orders is not the only factor that we need to consider when reviewing system results; a crucial factor is to have the data on which the system is tested be representative of all the different modes the market can adopt. The current system report has the results for 3 months, which is a period that might not represent all the different market modes.

Suggestions for Improvement

We can adjust this system in two ways.

First, we will want to define Data1 as longer or shorter bars, depending on how fast we can trade. If we are using smaller bars (i.e., 5-minute or 2-minute bars), we may want to modify the EntryOnSwing conditions to buy or sell at the highest High of the last 5 or 10 bars to avoid entering the market because of noise in the smaller data compression.

Second, we can define a method other than the fast and slow exponential moving average for determining the longer trend of the market. We can use any other analysis technique for this purpose, basing it on Data2.

Determining Short-Term Market Outlook

Trading systems can have an “alternate” use—they can help you determine short-term market outlook through statistics on how the market has reacted under certain circumstances. You can use this information, for example, to bias your trading towards bullish or bearish signals or as a filter for other systems.

One of the most simple systems of this nature is buy at the open of the first day of the week and exit from the position at the end of the day. By creating a trading system using this criteria, you will be able to determine what percentage of the first days of the week have been up days, the average points that it went up, and so on. We created a system using this criteria and called it

STAD2: Stats #2. Figure 7 shows the System Report for the Stats #2 System applied to a daily chart of the Dow Industrials:

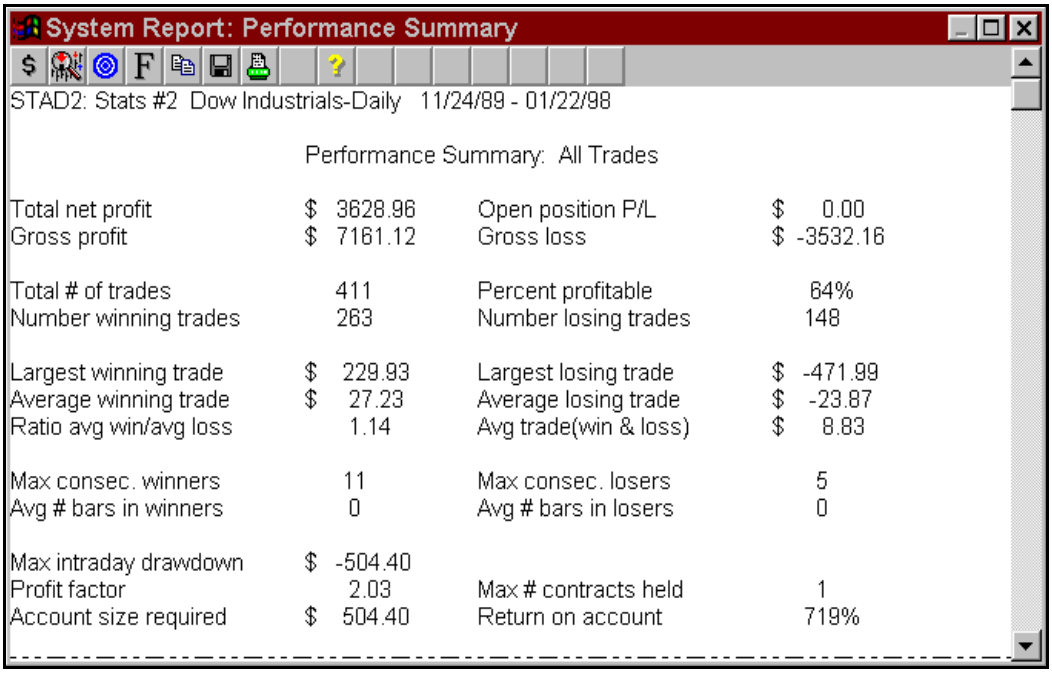


Figure 7. The System Report for the Stats #2 System applied to a daily DJIA chart

We can see that during the last 10 years, over 65% of Mondays have been up days, averaging better than 27 points. When we have had a down Monday, it has averaged a little over 24 points. Very similar results can be found for the S&P 500 Stock Index.

Now we can use this information as part of another trading strategy, perhaps as a filter or as an aid in defining the best entry point for long position, or we could try to improve on this simple concept and turn it into a trading system by itself.

For instance, we can see how different the system results are for Mondays while we are in a down trend than they are for Mondays while we are in a up trend. When analyzing this kind of system, don't limit your analysis to the average number of points per winning trade or the percentage of winners. Look at information like Maximum Draw Down and largest losing trades to see how these numbers are affected by variations on the original idea.

When testing these types of ideas, use one (or more) control groups in addition to your experimental group. For instance, for the DJIA buying Tuesdays, Wednesdays, Thursdays or Fridays all gave about 50% winning trades (each individually). This would lead you to believe that Mondays are statistically prone to be up days.

Another assumption we might want to investigate is that Friday the 13ths are down days. We can create a system that sells short at the open of every Friday the 13th and closes the position at the end of the day. We created a system called STAD2: Stats #3 with this criteria. Figure 8 shows the System Report for this system applied to the same DJIA. Notice that the Dow had a little under 50% of profitable trades, but the average profitable trade was nearly twice as large as the average losing trade.

System Report: Performance Summary

STAD2: Stats #3 Dow Industrials-Daily 01/05/89 - 03/20/98

Performance Summary: All Trades

Total net profit	$ 120.46	Open position P/L	$ 0.00
Gross profit	$ 320.27	Gross loss	$ -199.81
Total # of trades	15	Percent profitable	47%
Number winning trades	7	Number losing trades	8
Largest winning trade	$ 184.02	Largest losing trade	$ -59.29
Average winning trade	$ 45.75	Average losing trade	$ -24.98
Ratio avg win/avg loss	1.83	Avg trade(win & loss)	$ 8.03
Max consec. winners	3	Max consec. losers	6
Avg # bars in winners	0	Avg # bars in losers	0
Max intraday drawdown	$ -240.91		
Profit factor	1.60	Max # contracts held	1
Account size required	$ 240.91	Return on account	50%

Figure 8. The System Report for the Stats #3 System on the DJIA

Continuing with the idea of maintaining a control group, we can compare these results versus selling short every Friday. Selling short every Friday gave us a slightly better percentage, but the average win versus loss ratio was significantly affected. And when selling short on each of the days of the week for both the S&P 500 stock index and the Dow Industrials, Thursday appears to be the 'most negative day' (the best day to initiate short positions), with a little better than 50% of down days and the best average win/loss ratio, yet still not better than Friday the 13ths.

With these ideas, we can start to develop more refined systems that will give us valuable information for our trading. An example would be buying on Friday at the highest High of the week, staying in the market on Monday and exiting Tuesday at the Open. We wrote this simple system and called it STAD2: Stats #1. Figure 9 shows the System Report for the Stats #1 System on the S&P 500 stock index:

System Report: Performance Summary

STAD2: Stats #1 S&P 500 Stock Index-Daily 01/05/89 - 03/20/98

Performance Summary: All Trades

Total net profit	$ 111.14	Open position P/L	$ 2.34
Gross profit	$ 388.38	Gross loss	$ -277.24
Total # of trades	150	Percent profitable	61%
Number winning trades	92	Number losing trades	58
Largest winning trade	$ 20.48	Largest losing trade	$ -28.94
Average winning trade	$ 4.22	Average losing trade	$ -4.78
Ratio avg win/avg loss	0.88	Avg trade(win & loss)	$ 0.74
Max consec. winners	10	Max consec. losers	6
Avg # bars in winners	2	Avg # bars in losers	2
Max intraday drawdown	$ -55.30		
Profit factor	1.40	Max # contracts held	1
Account size required	$ 55.30	Return on account	201%

Figure 9. The System Report for the Stats #1 System on the S&P stock index

You can see that buying and staying in the market through Monday (which approximately 60% of Mondays close up) and exiting at the open of Tuesday gives us some very interesting results.

The EasyLanguage instructions for these 3 systems is very simple and as such it is not listed or discussed in this manual. Feel free to open the systems in the PowerEditor and review the EasyLanguage instructions.

APPENDIX A

Feedback on Volume 1

Each volume of the System Trading and Development Club will provide feedback on the previous volume.

Feedback can include general comments from clients, or general omissions we made from Volume 1. It can also include discussion on one or more of the systems, including answers to questions clients have asked regarding the specific system. This appendix will also address any errors, if any, found in the systems from the previous volume.

Please feel free to send us your comments and/or questions on the current volume, and we will include them in the next volume. Send your comments via e-mail to:

stadclub@omegaresearch.com

Thank you for your continued support!

General Comments

Volume 1 of the System Trading and Development Club included a system called Pivot Points. This system idea was provided by Stanley Dash. We apologize for not giving due credit in the previous manual.

Moving Average Crossover System

A client inquired as to why the Moving Average Crossover System did not generate any trades when applied to the $OEX (S&P 500). The reason no trades were generated on an $OEX chart is as follows:

The MACO System determines entry prices by calculating a value of 3% above the highest High and 3% below the lowest Low. Please see STAD Club Volume 1, pages 16 - 17). When you apply the system to symbols that trade in the 1000s like the S&P 500, the resulting entry prices are too far away from the market action to be realistic entry prices. For these types of markets, 3 percent is an invalid number. A more useful percentage to use would be something closer to 1 percent. Likewise, if you applied the system to a symbol and were getting too many trades, you would have to increase the percentage.

We recommend that you use an input for the percent and then change this value according to the market you are trading. The system as we designed it does not use an input for the percent but you can easily include one.

Trend Line Automatic System

A client reported a problem with this system.

The Problem

The algorithm for determining the angle of the trendline and basing your trades on a slope greater than 10 does not work. Trades were generated or not generated regardless of this value.

The system was originally designed as follows (please see STAD Club Volume 1, page 33):

Long Entries

If the close of the current bar crosses over the down-trend line and this trend line has a slope greater than 10 degrees, then the system will place a buy order on the next bar at the open. RHTL-Ref is a variable that holds the ID number of the down-trend line. We set the value of this variable to -1, and it will hold this value until the first trend line is drawn. Therefore, before we obtain the trend line information and check for the cross over and slope, we will check RHTLRef to make sure it is not equal to -1:

```
If RHTLRef <> -1 then Begin
        BuyPrice = TL_GetValue( RHTLRef , Date, Time );
        PrevBuyPrice = TL_GetValue( RHTLRef , Date[1] , Time[1] );
        If Close Crosses Over BuyPrice AND ArcTangent( PrevBuyPrice / BuyPrice) >10
then
                Buy next bar at Market;
End;
```

Short Entries

If the close of the current bar crosses under the up-trend line and this trend line has a slope greater than 10 degrees, then the system will place a sell order on the next bar at the open. As with the long entry, we use the variable RLTLRef to make sure a trend line is drawn before we obtain the trend line information and check for both the cross over and slope:

```
If RLTLRef <> -1 then Begin
        SellPrice = TL_GetValue( RLTLRef , Date , Time );
        PrevSellPrice = TL_Getvalue( RLTLRef , Date[1] , Time[1] );
        If Close crosses under SellPrice AND ArcTangent(SellPrice / PrevSellPrice)>10
then
                Sell next bar at Market;
End;
```

Analysis and Correction

The portions that are in bold were supposed to determine when the slope of the trendline was greater than 10. However, we realized that the algorithm to determine the slope is more involved than simply using the ArcTangent function. We therefore determined that instead of looking at the slope of the trendline, a better way to accomplish our goal is to look at the number of points the trendline has increased per bar. This is a much simpler calculation. The resulting instructions are:

Long Entries

```
If RHTLRef <> -1 then Begin
        BuyPrice = TL_GetValue( RHTLRef , Date, Time );
        PrevBuyPrice = TL_GetValue( RHTLRef , Date[1] , Time[1] );
        If Close Crosses Over BuyPrice AND ABSValue( BuyPrice - PrevBuyPrice) > 2 then
                Buy next bar at Market;
End;
```

Short Entries

```
If RLTLRef <> -1 then Begin
        SellPrice = TL_GetValue( RLTLRef , Date , Time );
        PrevSellPrice = TL_Getvalue( RLTLRef , Date[1] , Time[1] );
        If Close crosses under SellPrice AND ABSValue(SellPrice - PrevSellPrice) > 2 then
                Sell next bar at Market;
End;
```

We used the Absolute Value function (ABSValue) to determine the number of points the trendline has increased per bar. When the trendline is horizontal, the resulting value is zero, so the greater the increase, the greater the resulting value.

INDEX

T

V

W