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 3

Welcome to Volume 3 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 third 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 2. Those wanting additional discussion on the systems will benefit from this appendix.

The systems presented in this volume incorporate popular concepts such as Bollinger Bands, volatility, key reversals, DMI and variable zones. 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.*

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 **MAYJUN98.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 STAD3. 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 4 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 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

E-Mail Address:

easylang@omegaresearch.com

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

Two of the most frequently-asked questions among system developers are, “How do I know if this system is tradable? How can I know if the system will work for me?”

And the reason so many traders keep asking themselves these questions is because there is no straight answer, there is no one number you can look at in the results of a system to answer these questions. There are some numbers that are more important than the others, but there is no one single number. And actually, evaluating a system is a complicated task and you should give it the time and effort it deserves.

The purpose of this chapter is to provide you with some insight into system evaluation and discuss the numbers in TradeStation’s System Report that we think are most important to look at when evaluating trading systems. You need to fully understand a system and its behavior before you should trade it. We also discuss some of the enhanced system testing functionality found in Portfolio Maximizer 4, the portfolio analysis software also available from Omega Research.

The information in this chapter will help you better determine if you can and should trade a particular system.

In This Chapter

Understanding the System Report

The System Report is the single most important tool in analyzing the performance of a system and understanding its characteristics. This section covers the fields we think are most important. You should understand all the fields and ultimately decide for yourself exactly how to evaluate a system. Figure 1shows a sample System Report.

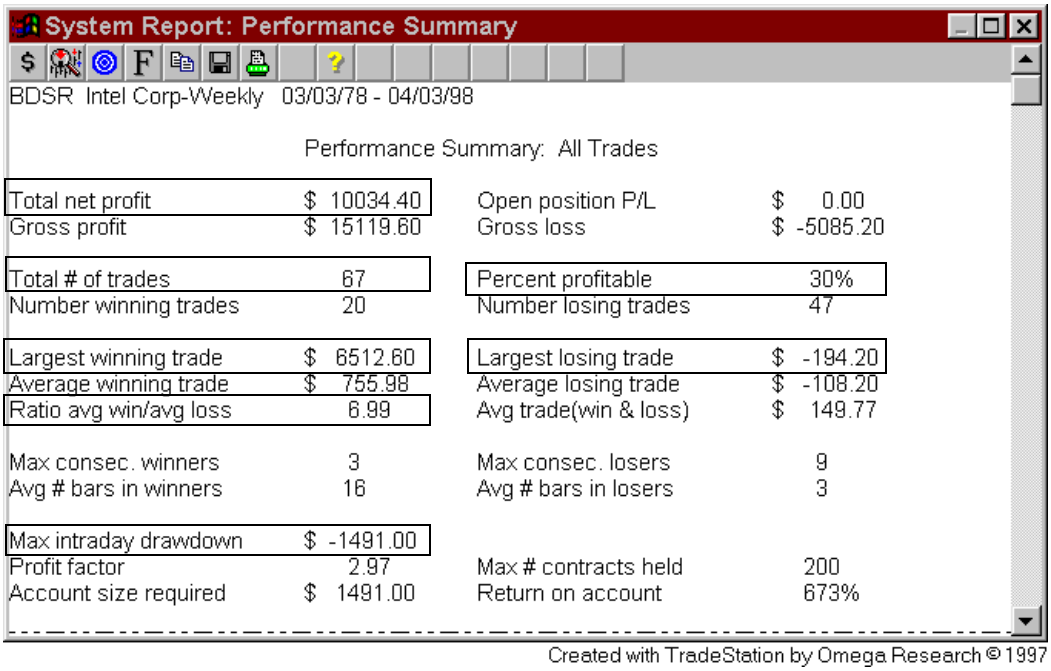


Figure 1. Sample System Report

Total Net Profit

This field is very important. We are all studying technical analysis and system testing in order to make profits, to grow our capital, so the main objective of any system is to have as big a total net profit as possible. However, when developing and fine tuning our systems or even backtesting on historical data, you cannot look at this figure by itself—it means very little without the rest of the analysis.

Percent Profitable

The higher this percentage the better; it is a very important number when gauging the psychological impact of a system. It lets you know how many trades out of your total number will be profitable. An extremely low or high percent of profitable trades will not break or make a system by itself—a system that has a great average winning versus losing trade ratio can afford to have a low percent of winning trades, as the few big winning trades will cover the losers (this is one of the characteristics of trending systems).

Again, this number is important when gauging the psychological impact of a system. Some systems that are profitable might have large winning trades that are few and far between, and this may not suit a trader's need for positive reinforcement.

Ratio Avg Win/Avg Loss

This number is the ratio of average winning trades versus average losing trades. Similar to Percent Profitable, the higher the ratio the better, but a very low or very high number does not necessarily make a system good or bad. If a system has a very low average of winning versus losing trades ratio but an extremely high percent of winning trades, it will still be profitable.

The many small profitable trades will more than cover the infrequent, but bigger, losing trades (this is one of the characteristics of volatility expansion systems). These last two numbers should always be looked at side by side when studying system results.

Maximum Intra-day Drawdown

This number tells you what the maximum equity dip was during the time period represented in the chart. Many traders argue that a system is good if the maximum intra-day drawdown is under 20% of the total net profit, but drawdown, regardless of how small it is, can make a system untradable because of what it represents with respect to the size of the account used to trade the system.

To put it in very simple terms, say you currently have $1,000 in your checking account, and that during the month you will receive $500 in payment and pay out $1,300 in expenses. The payment comes at the middle of the month. Even though your monthly balance is positive, you still may not make it to the end of the month without running out of money. For example, if during the first week of the month you pay $600 in rent, $300 for your car and insurance, and $100 for food, and then you need to pay $20 for gas to start week two, you won't have the money for gas; your drawdown was $1,020 before you were able to collect the $500.

Total # of Trades

This number, although many disregard it, represents the cost of the system. A system that is extremely active will require much more attention and follow up (and much more in commissions). If a system that is put on a daily chart going back one year produces 100 trades, it means that it produced a little over two trades per week. This might be too frequent for someone trading part time but too slow for someone trading full time.

Largest Winning Trade

The largest winning trade should NOT represent a significant percent of the total net profit; the effect of over-optimizing (or curve fitting) your systems is usually expressed as a very large winning trade. You need to be comfortable with the results after removing the largest winning trade from the System Report.

For example, evaluate the total net profit after subtracting from it the largest winning trade and look at the percentage the drawdown represents of the total net profit after taking out the largest winning trade. You always want the largest winning trade to be a number as close as possible to the average winning trade—the closer these two numbers are the more stable the system will prove to be.

This is the rule of thumb for any system, but keep in mind that when you're dealing with trending systems, which are designed to cash in on the big moves, a few large winning trades will probably make most of the profits.

Largest Losing Trade

While the largest winning trade is the most irrelevant trade, the largest losing trade is the single most important trade of the entire simulation. It invariably points to the weakness of your system. The first step to take when improving any system is to find the largest losing trade and then try to determine why the system did not protect you against this loss, and how you can make the system avoid a trade similar to this one in the future.

You always want the largest losing trade to be a number as close as possible to the average losing trade—the closer these two numbers are the more stable the system will prove to be.

Using the System Equity Curve

A significant element when studying any trading system is the shape of the equity curve. If the system is stable, its equity curve will be a steady up sloping line. Again, one effect of curve fitting your systems is one or few very large winning trades, and this can be seen easily by looking at the shape of the equity curve. An equity curve that is very volatile (has many spikes and sudden up and down movements) yet ends with a substantial profit is also a sign of an over-optimized and/or untradable system.

A smooth rising equity curve will be a sure sign of a system that is consistently profitable, that it has small drawdown (remember the maximum intra-day drawdown is defined as the largest equity dip), and that is easier to trade. Remember that your trading account balance will have a striking resemblance to the equity line of the systems you trade.

To view the equity curve of any system applied to a chart in TradeStation (Version 4 or higher), apply the System Equity Indicator to the chart containing the system. Figure 2 shows a chart with the System Equity Indicator applied.

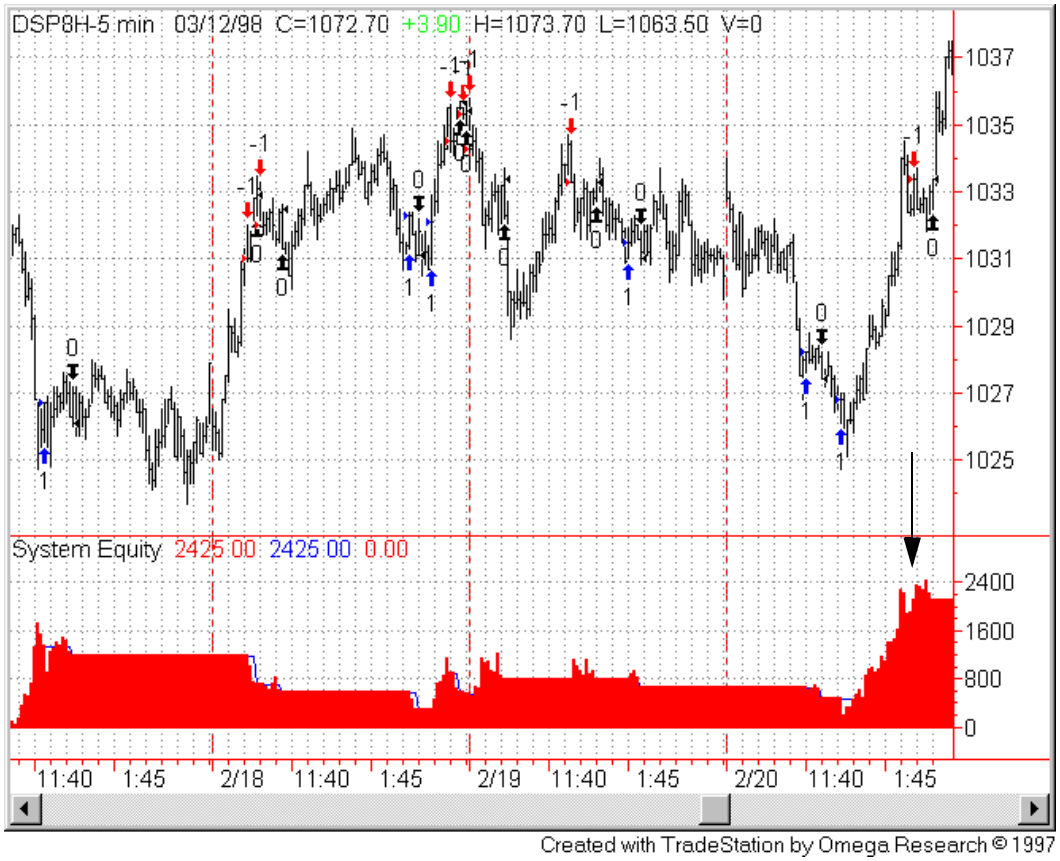


Figure 2. Chart with the System Equity Indicator applied

As you can see in Figure 2, in spite of having over $2,000 of profit in a few trading days, the system's equity curve is quite volatile, and you can see how most of the profits were made by one trade toward the middle of the last day on the chart. What if you hadn't traded that day?

Using Portfolio Maximizer for TradeStation

Portfolio Maximizer for TradeStation gives you the power to create and historically test groups of trading systems and securities combined in portfolios, revealing trading combinations that are stronger and more profitable than individual systems or markets alone. Also, Portfolio Maximizer provides an extensive performance report that opens the door to a host of new ways to gauge your portfolio's true performance. It provides an individual report on one system as well as a portfolio report. This section describes the more important fields on the individual report.

Standard Deviation of Avg Winning Trades/Std Dev of Avg Losing Trades

One of the most desirable traits of a trading system is predictability. The reason we backtest our systems is to know how they behaved in the past in order to know what to expect from them in the future. One way to statistically determine stability is by calculating the standard deviation of the trades from the average. When we calculate the average winning trade and determine the standard deviation of the trades from the average, we can then state that approximately 96% of the trades will fall between the average plus or minus the standard deviation. For example, if the average winning trade gives us $500 profit, and the standard deviation is 100, we know that most of the winning trades (about 96%) will give us profits between $400 and $600. If the standard deviation is 490, we know that the trades will most likely fall between $10 and $990.

Once you begin looking at this number you will notice that invariably the equity curve smooths out as the standard deviation of winning and losing trades decreases.

Time Analysis

A figure as important as any other when testing your systems is how much time the system has your capital in the market. A system can provide great results, but it might require significant capital to be dedicated exclusively to it. There are several reasons why you want to study time when looking at system results:

1. **Risk**. The more time the system has money in the market, the more the capital is exposed to violent moves in the market, thus higher risk.

2. **Availability of funds**. If system A can provide you with a 20% annual return, and system B can provide you with 30% annual return, yet A is in the market 40% of the time versus 75% for system B, this means that you will have your capital at your disposal for seven months out of the year versus only three for system B. In those four months (difference between spare time of system A and B), you could have been investing your capital elsewhere.

Frequency of Trade

You should always keep an eye on how frequently the system trades because it will have an effect on how much of your time you will spend following the system and how.

Maximum Adverse Excursion

Maximum Adverse Excursion (MAE) is an extremely useful tool that enables you to visually see how your system is handling drawdown and where you can cut your losses short. The graph uses dots to represent trades (up triangles for winners, down triangles for losers) and places one dot on the graph for each trade. The profit/loss of the trade is on the vertical scale, and the maximum trade drawdown is on the horizontal scale.

By looking at the graph, you will be able to see at exactly what drawdown the trades start becoming losers. If virtually all the trades that had a drawdown over $1,000 turned out to be losing trades, you can safely assume that placing a stop loss at $1,000 will reduce your drawdown and improve your profits (by reducing your gross losses) without affecting any winning trades. You don't want your stop loss orders (money management) to affect your winning trades.

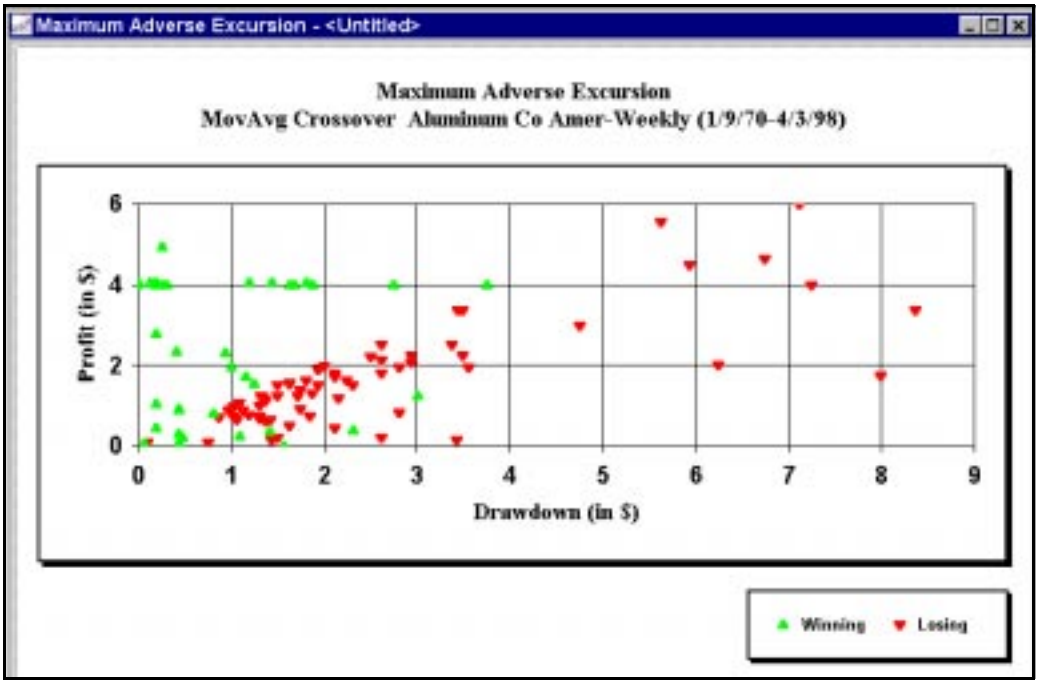


Figure 3. Maximum Adverse Excursion graph showing winning and losing trades and their drawdown

Figure 3 shows the Maximum Adverse Excursion graph. Notice that after a drawdown of 3 points the majority of trades were losers. Placing a stop loss at that point should substantially help reduce drawdown and gross losses.

Maximum Favorable Excursion

When developing a system, it is as important to time your exits as it is to time your entries. Maximum Favorable Excursion (MFE) is a graph that will allow you to see if your exits are letting the positions run too long before capturing profits. It is a scatter chart very similar to Maximum Adverse Excursion, showing all the trades (up triangles for winners, down triangles for losers) with the profit/loss as the vertical axis and the run-up (maximum rise in equity, which is opposite of drawdown) as the horizontal axis. Figure 4 shows a sample MFE graph.

One of the things to look for is a concentration of losing trades that had significant run-up. If a significant number of losing trades had run-up, this would be an indication that the system is timing the entry correctly (since the trade was making a significant profit at some point during

its lifetime), and the system failed to capture these profits, letting the trades become losing trades.

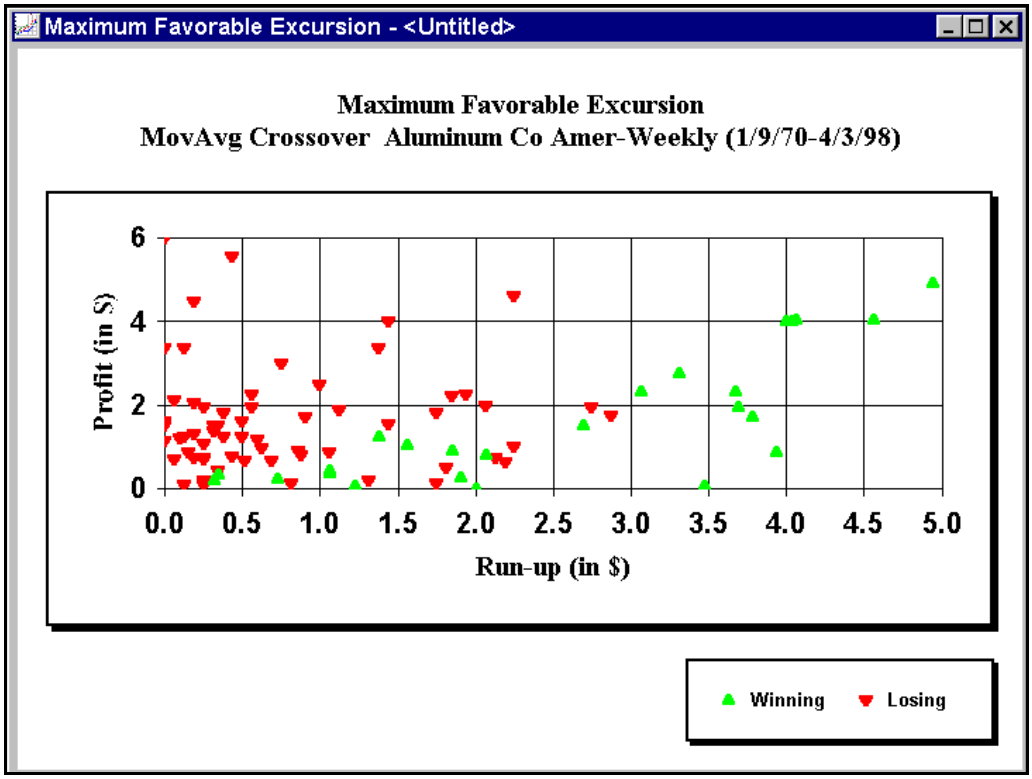


Figure 4. Maximum Favorable Excursion graph, showing winning and losing trades and their run-up

Notice that there are some losing trades that have significant run-up, say over 2 points. We could look at these and make an attempt to capture more profits.

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 that are all designed to capture big moves and limit losses during directionless and/or volatile phases.

In This Chapter

Buy Dips/Sell Rallies

The idea behind the Buy Dips/Sell Rallies trading system is to buy after a countertrend decline in an uptrend but only once the uptrend resumes, and likewise, to sell short after a countertrend rally in a downtrend but only once the downtrend resumes. It is commonly accepted that trending markets will experience one or more corrections as they continue along the trend. We want to be able to identify a trend but then find the best possible point to enter the market in order to take advantage of the trend. Our thinking is that we will identify a trend but wait until a countertrend occurs. Then, if the trend continues after the countertrend decline or rally, indicating that the trend is strong and likely to continue for an acceptable period, we will enter the market.

First, we needed a way to determine the existence and direction of the trend. Most systems use moving averages to determine the direction of the trend, so we wanted to use something a little different. We decided to use the Relative Strength Index (RSI) Indicator. The RSI is an oscillator generally used to find overbought and oversold conditions and to show divergence from the price. Most commonly, markets with an RSI over 70 or 80 are considered to be overbought, and markets with an RSI below 20 or 30 are considered to be oversold. In addition, the value of 50 can serve the same purpose as the zero line in other oscillators. A crossing above or below 50 can signal the slowing down of a current trend or a trend reversal.

In Buy Dips/Sell Rallies, we will use the RSI and this 50 level to determine the overall direction of the market. We will assume that when the RSI is greater than 50, the market is in an uptrend, and when the RSI is less than 50, the market is in a downtrend.

We also needed to incorporate a way to identify countertrend price swings. We decided to use the Directional Movement Index (DMI) spread. The traditional DMI Indicator consists of three values: the DMI Plus, the DMI Minus, and the ADX. For the purpose of our analysis, we will use the DMI spread, in other words, the difference between DMI Plus and DMI Minus. DMI Plus measures upward price movement and DMI Minus measures downward price movement. When the difference between them is less than zero in an uptrend, or greater than zero in a downtrend, we'll assume a countertrend price swing

The key is to use a longer period for the calculation of the RSI than for the calculation of the DMI spread. For this system, we will use 20 bars to calculate the RSI and 5 to calculate the DMI values. This enables us to use the RSI to determine the overall trend and the DMI to identify shorter-term corrections within the overall trend.

Therefore, we wrote our system so that when the RSI is greater then 50 and the DMI spread is less than zero, we have a buy setup. After the buy setup, we wait for the DMI spread to cross above zero. When this happens, we obtain the high price of the current bar and place a buy stop at the high. This way, to enter the market, price movement has to confirm our setup.

Likewise, when the RSI is less than 50 and the DMI spread is greater than zero, we have a sell setup. After the sell setup, we wait for the DMI spread to cross below zero. When this happens, we obtain the low price of the current bar and place a sell stop at the low. Again, by using a sell stop, price movement has to confirm our setup in order for us to enter the market.

The buy and sell orders remain active until the market conditions reverse, i.e., the RSI or DMI spread crosses back over its respective limit. They remain active with the original entry price, the high or low price of the bar on which the buy or sell setup conditions first became true.

Once we determined how we were going to enter the market, we had to decide how to exit. We'll use a standard trending system exit, as follows: once we enter the market on the buy stop, we'll calculate the 4-bar average of the range and subtract this value from the low. This will be our long exit price for bar 2. On the subsequent bar, we'll 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 us our exit price for bar 3 and beyond.

Likewise, once we enter the market on the sell stop, we'll calculate the 4-bar average of the range and add this value to the high. This will be our short exit price for bar 2. On the subsequent

bar, we'll get the exit price calculated in the last point and subtract from it a third of the difference between the previous stop price and the high. Repeating this operation at the end of every bar will give us the exit price for bar 3 and beyond.

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

Figure 1 shows a chart with the Buy Dips/Sell Rallies system applied. We also applied the RSI w/o Zones Indicator, which is built in to TradeStation, and the DMISpread Indicator, which we wrote in EasyLanguage and is included on your STAD Club CD (STAD3: DMISpread). The DMISpread Indicator plots a histogram showing the relationship between the DMI Plus and the DMI Minus.

When you write a system that uses the same calculation as an indicator and then apply the indicator along with the system to a chart, make sure the inputs for the indicator are set to the same values as the inputs in the system.

If they are not the same, the indicator plot will not correspond to the system arrows on your screen. This is an easy oversight to make.

Figure 1. The Buy Dips/Sell Rallies trading system applied to a chart, along with the RSI and DMI indicators

Notice that the chart in Figure 1 shows only long entries and exits. For the period shown, the RSI never fell under 50, which has to happen to satisfy the sell criteria.

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 RSI and the DMI spread. The setup, entries and exits are described next:

Setup

a) Calculate the RSI value and compare it to 50 to determine the overall direction of the market.

b) Calculate the DMI spread and compare it to zero to identify a countertrend decline or rally.

Long Entries

a) When the buy setup criteria are met (RSI is greater than 50 and DMI is less than zero) and then the DMI spread crosses above zero, place a buy stop at today's high.

Short Entries

a) When the sell setup criteria are met (RSI is less than 50 and DMI is greater than zero) and then the DMI spread crosses below zero, place a sell stop at today's low.

Exit Orders

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

b) On the subsequent 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 gives us our exit price for bar 3 and beyond.

c) Once the bar of entry of a short position is closed, calculate the 4-bar average of the range and add this value to the high. This is our short exit price for bar 2.

d) On the subsequent bar, we'll get the exit price calculated in the last point and subtract from it a third of the difference between the previous stop price and the high. Repeating this operation at the end of every bar gives us the exit price for bar 3 and beyond.

e) Place a money management stop (stop loss), the dollar amount of which will depend on what and how many shares/contracts you 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: Buy Dips/Sell Rallies (STAD3: BuyD SellR)

```
Input: Price(Close), RSILen(20), RSILevel(50), DMILen(5);

Vars: UpTrend(False), DnTrend(False), DMISpread(0), BuySetup(False), SellSetup(False),
LongEntryPrice(0), ShortEntryPrice(0);

{ Determining the buy and sell setup }
UpTrend = RSI(Price, RSILen) >= RSILevel;
DnTrend = RSI(Price, RSILEn) < RSILevel;

DMISpread = DMIPlus(DMILen) - DMIMinus(DMILen);

If UpTrend and DMISpread < 0 then Begin
        BuySetup = True;
        SellSetup = False;
End;
```

```
If DNTrend and DMISpread > 0 then Begin
        SellSetup = True;
        BuySetup = False;
End;

{ Long entry orders }
If BuySetup then Begin
        If BuySetup[1] = False then LongEntryprice = High;
        Buy next bar at LongEntryPrice Stop;
End;

{ Short entry orders }
If SellSetup = True then Begin
        If SellSetup[1] = False then ShortEntryPrice = Low;
        Sell next bar at ShortEntryprice Stop;
End;

{ Exit orders }
MP = MarketPosition;
If MP = 1 and MP[1] <> 1 thenStopPrice = Low - Average(Range,4);
If MP = -1 and MP[1] <> -1 thenStopPrice = High + Average(Range,4);
If MP = 1 then Begin
        Exitlong next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
If MP = -1 then Begin
        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	Description
Price	Close	Price used to calculate the RSI.
RSILen	20	Length, expressed in bars, used to calculate the RSI.
RSILevel	50	Value under which the RSI must fall for the market to be considered in a downtrend, and over which it must rise for an uptrend.
DMILen	5	Length, expressed in bars, to use to calculate the DMI Plus and DMI Minus values.

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

```
Vars: UpTrend(False), DnTrend(False), DMISpread(0), BuySetup(False), SellSetup(False),
LongEntryPrice(0), ShortEntryPrice(0);
```

We first calculate the RSI and compare it to 50, storing the values in the variables UpTrend and DnTrend. We also calculate the DMI spread and store the value in the variable DMISpread.

```
UpTrend = RSI(Price, RSILen) >= RSILevel;
DnTrend = RSI(Price, RSILEn) < RSILevel;
DMISpread = DMIPlus(DMILen) - DMIMinus(DMILen);
```

Then, we look at the values in the UpTrend and DnTrend variables and evaluate the DMI spread. When the RSI is greater than 50 (UpTrend = True) and DMISpread is less than zero, BuySetup is set to true and SellSetup is set to false. When RSI is less than 50 (DnTrend = True) and DMISpread is greater than zero, SellSetup is set to true and BuySetup is set to false.

```
If UpTrend and DMISpread < 0 then Begin
    BuySetup = True;
    SellSetup = False;
End;

If DNTrend and DMISpread > 0 then Begin
    SellSetup = True;
    BuySetup = False;
End;
```

Placing the results of the evaluations in the SellSetup and BuySetup variables enables us subsequently to use them in our IF-THEN statements in order to place our buy and sell stop orders.

Long Entries

When the variable BuySetup is true, which means the RSI is greater than 50 and the DMI Spread is less than zero, we check to make sure that the value of the variable BuySetup on the previous bar was false. If it was, we set the variable LongEntryPrice to the high of the current bar. We want to make sure that we set the entry price to the high of the bar on which the buy setup conditions first became true. We hold on to this price until the order is generated or until conditions reverse.

We also place a buy stop order for the next bar at the high of the current bar.

```
If BuySetup then Begin
    If BuySetup[1] = False then LongEntryprice = High;
    Buy next bar at LongEntryPrice Stop;
End;
```

Short Entries

When the variable SellSetup is true, which means the RSI is less than 50 and the DMI Spread is greater than zero, we check to make sure that the value of the variable SellSetup on the previous bar was false. If it was, we set the variable ShortEntryPrice to the low of the current bar. We want to make sure that we set the entry price to the low of the bar on which the sell setup conditions first became true. We hold on to this price until the order is generated or until conditions reverse.

We also place a sell stop order for the next bar at the low of the current bar.

```
If SellSetup = True then Begin
    If SellSetup[1] = False then ShortEntryPrice = Low;
    Sell next bar at ShortEntryprice Stop;
End;
```

Exit Orders

Market conditions could be such that the system will reverse without exiting; however, because of the way in which we designed the system, it is highly unlikely that the system will reverse. Instead, it will exit from a long or short position when the exit price is met.

First, we obtain our market position using the MarketPosition function. Then, on the first bar after we establish our long position, we set the variable StopPrice to the low of the current bar minus the 4-bar average of the range. Likewise, on the first bar after we establish our short position, we set the variable StopPrice to the high of the current bar plus the 4-bar average of the range.

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

Then, when in a long position, we place an stop order to exit our long position at the StopPrice or better, and reset our StopPrice to the original StopPrice plus a third of the difference between the low of the current bar and the original StopPrice. Similarly, when in a short position, we place a stop order to exit our short position at the StopPrice or better, and reset our StopPrice to the original StopPrice minus a third of the difference between the original StopPrice and the high of the current bar.

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

If MP = -1 then Begin
      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 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.*

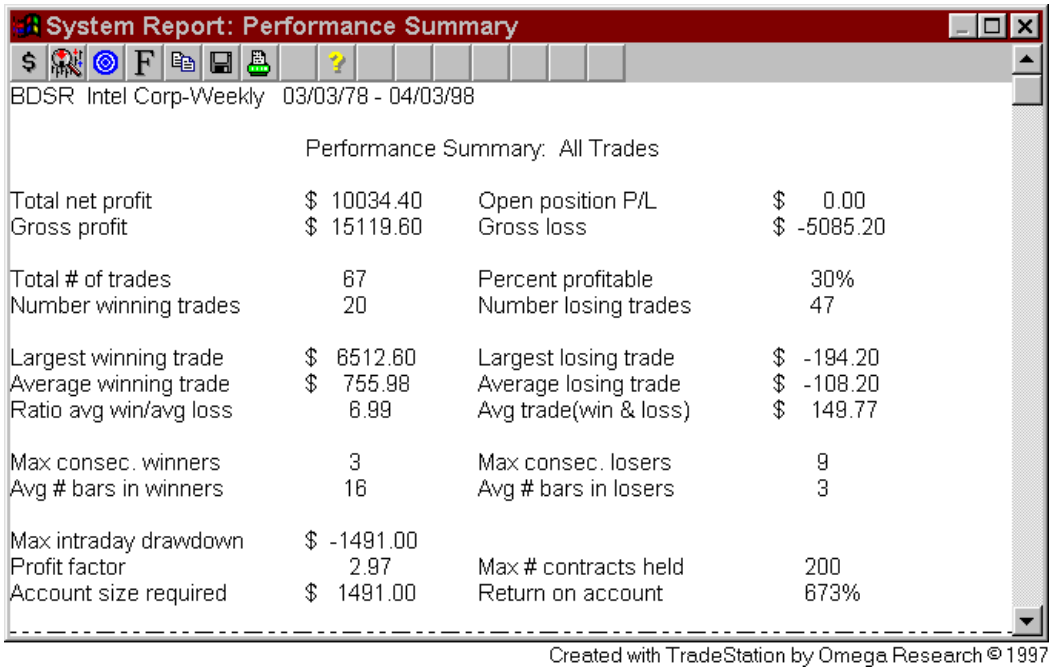
b) Under the **Stops** tab, we enabled a money management stop (the **Money Mngmnt** check box) and entered an appropriate dollar amount in the edit box. This option holds the dollar amount per position or dollar amount per contract/share we want to risk before exiting out of the position.

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 Buy Dips/Sell Rallies trading system.



System Report: Performance Summary

BDSR Intel Corp-Weekly 03/03/78 - 04/03/98

Performance Summary: All Trades

Total net profit	$ 10034.40	Open position P/L	$ 0.00
Gross profit	$ 15119.60	Gross loss	$ -5085.20
Total # of trades	67	Percent profitable	30%
Number winning trades	20	Number losing trades	47
Largest winning trade	$ 6512.60	Largest losing trade	$ -194.20
Average winning trade	$ 755.98	Average losing trade	$ -108.20
Ratio avg win/avg loss	6.99	Avg trade(win & loss)	$ 149.77
Max consec. winners	3	Max consec. losers	9
Avg # bars in winners	16	Avg # bars in losers	3
Max intraday drawdown	$ -1491.00		
Profit factor	2.97	Max # contracts held	200
Account size required	$ 1491.00	Return on account	673%

Created with TradeStation by Omega Research © 1997

Figure 2. Sample System Report for the Buy Dips/Sell Rallies trading system

On the surface, this system returns some great numbers, for example, the impressive ratio of winning vs. losing trades (6.99). Also, 30 as the percent of profitable trades is to be expected since the system is a trending system that looks to enter the trade once there is a correction. But let's look a little deeper.

First, the largest wining trade is approximately 60% of the total net profit. This is not necessarily a bad statistic, since a trending system is designed to capture and make the most profit from the big moves. However, if we look at the equity curve of the system, we'll notice that it remains around the break even point, and it's only the last few trades that give the system decent results.

However, as you can see from the chart in Figure 3, which shows the system along with the System Equity Indicator, the stock wasn't showing major trends up until mid-1995 where it made a major up move that the system capitalized on.

Figure 3. The Buy Dips/Sell Rallies trading system and System Equity Indicator applied to a weekly chart of Intel

Overall, this system seems to behave as expected, capitalizing on trends and cutting losses short when the trend goes against its positions. We need to test it on additional data to determine its robustness.

Suggestions for Improvement

Even though the system appears to recognize bad trades early and exit with minimal losses, the 30/70 profit/loss ratio can be discouraging and makes the system psychologically difficult to trade. It's difficult to withstand losing trades 70% of the time!

To improve the system, we may want to add to our criteria such that we enter the market not only when the DMI spread crosses the zero line, but also when the DMI spread ticks up from below zero (meaning that the DMI spread is greater than the DMI spread yesterday) or ticks down from above zero (meaning that the DMI spread is less than the DMI spread yesteday).

We could also use other filters, such as the ADX, to detect when the market is in a stronger long-term trend, or use historical fundamental information, such as the EPS reports available through Dial Data, and design the system to enter trades during minor corrections in the overall trend.

Bullish & Bearish Engulfing Patterns

In a Japanese Candlestick chart, the body, or real body as it is called, is defined by the distance between the open and the close of the bar. Also, it will be hollow when the close is greater than the open, and filled when the close is less than the open. Figure 4 illustrates this.

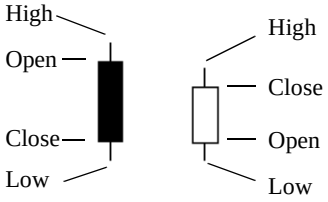


Figure 4. Japanese Candlesticks

Japanese Candlestick chartists look at not only the individual candlesticks but also at the patterns and formations that result from a group of candlesticks in an attempt to determine market action. One interesting pattern is the engulfing pattern. A Bullish Engulfing pattern is defined as a large, hollow real body that engulfs a small, filled real body in a downtrend, and a Bearish Engulfing pattern is defined as a large, filled real body that engulfs a small, hollow real body in an uptrend.

This pattern is interesting because it is considered to be an indication of a reversal. For example, it can be argued that if the market is in an uptrend and you find a Bearish Engulfing pattern, a trend reversal is likely, and vice versa. We decided to build a system using these patterns to identify trend reversals.

First we had to decide what to use to identify whether or not the market is trending, and in what direction. We decided to use an exponential moving average calculation. We'll say that the market is in an uptrend when the 10-day exponential average of the close is greater than the same average one bar ago, and we'll say the market is in a downtrend when the 10-day exponential average of the close is less than the same average one bar ago.

Then, we had to define the specifications for the patterns. We'll say that a real body is large when it is at least 125% the size of the 10-day average real body, and we'll say it's small when it is no more than 75% the size of the 10-day average real body.

The chart in Figure 5 shows an example of a Bullish Engulfing pattern. We also applied the Moving Average-Exponential Indicator plotting the 10-day exponential average of the closes.

Keep in mind that the patterns are found through mathematical evaluation of the price data; you don't have to apply the system to a Candlestick chart.

You can apply this system to an OHLC bar chart or any other type of chart and TradeStation will still identify the patterns.

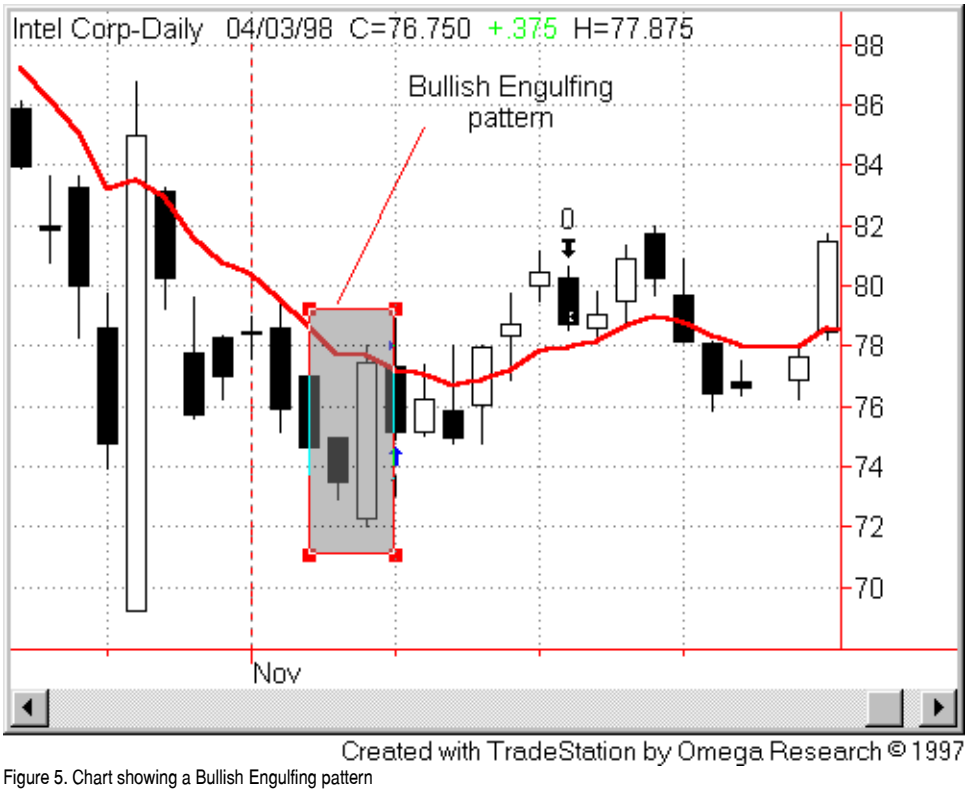


Figure 5. Chart showing a Bullish Engulfing pattern

To enter a long position once we find a Bullish Engulfing pattern, we place a buy stop at the high of the day on which the pattern occurred. The buy signal will remain active for 3 bars after the Bullish Engulfing pattern. When filled, we'll set a trailing stop at the highest close since the long position was initiated minus half the average range of the last 4 bars.

To enter a short position once we find a Bearish Engulfing pattern, we place a sell stop at the low of the day on which the pattern occurred. The sell signal will remain active for 3 bars after the Bearish Engulfing pattern. When filled, we'll set a trailing stop at the lowest close since the short position was initiated plus half the average range of the last 4 bars. Figure 6 shows the Bullish & Bearish Engulfing Patterns trading system applied to a chart.

Figure 6. The Bullish & Bearish Engulfing Patterns system applied to a daily Intel chart along with the Moving Average-Exponential Indicator

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 identifying the Bearish and Bullish Engulfing patterns. The setup, entry and exits are described next.

Setup

a) Calculate a 10-bar exponential average of the close and determine if the market is in an uptrend or in a downtrend by comparing the current bar's value to the previous bar's value.

b) Calculate the average real body size of the last 10 candlesticks.

c) Search for Bearish and Bullish Engulfing patterns.

d) When the market is in a downtrend and a Bullish Engulfing pattern is found, capture the high for use as the long entry price. Likewise, when the market is in an uptrend and a Bearish Engulfing pattern is found, capture the low for use as the short entry price.

Long Entries

a) If a Bullish Engulfing pattern has occurred during an downtrend in the last 3 bars, and we are not currently in a long position, place a buy stop order at the high price captured during the setup.

Short Entries

a) If a Bearish Engulfing pattern has occurred in an uptrend the last 3 bars, and we are not currently in a short position, place a sell stop order at the low price captured during the setup.

Exits

a) Once our buy stop order is filled, set a trailing stop at the highest close since the long position was initiated minus half the average range of the last 4 bars.

b) Once our sell stop order is filled, set a trailing stop at the lowest close since the short position was initiated plus half the average range of the last 4 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: Bearish & Bullish Engulfing Patterns (STAD3: BB Eng Patt)

```
Input: Price(Close), Length(10), BodyLen(10);
Vars: BearEngulfing(False), BullEngulfing(False), LongEntryPrice(0), ShortEntryPrice(0),
UpTrend(False), DownTrend(False), AvgBody(0), HighestC(0), LowestC(999999);

{ Setup calculations }
UpTrend = XAverage( Price, Length ) > XAverage( Price, Length )[1] ;
DownTrend = XAverage( Price, Length ) < XAverage( Price, Length )[1] ;
AvgBody = Average( AbsValue(Close - Open), BodyLen );

BullEngulfing = ( Close > Open  AND Close[1] < Open[1] ) AND
        ( Close > Open[1] AND Open < Close[1] ) AND ( Close - Open > AvgBody * 1.25 ) AND
        ( Open[1] - Close[1] < AvgBody * .75 );
BearEngulfing = ( Open > Close AND Open[1] < Close[1] ) AND ( Open > Close[1] AND
        Close < Open[1] ) AND ( Open - Close > AvgBody * 1.25 ) AND
        ( Close[1] - Open[1] < AvgBody * .75 );

If DownTrend and BullEngulfing thenLongEntryPrice = High;
If UpTrend and BearEngulfing thenShortEntryprice = Low;

{ Entry orders }
If MRO( DownTrend and BullEngulfing, 3 , 1) > -1 AND MarketPosition <> 1 then
        Buy next bar at LongEntryPrice Stop;
If MRO( UpTrend and BearEngulfing, 3 , 1) > -1 AND MarketPosition <> -1 then
        Sell next bar at ShortEntryPrice Stop;

{ Long exit order }
If MarketPosition = 1 then Begin
        If Close > HighestC then HighestC = Close;
        Exitlong at HighestC - Average( Range, 4 ) / 2 Stop;
End
Else HighestC = 0;

{ Short exit order }
If MarketPosition = -1 then Begin
        If Close < LowestC then LowestC = Close;
        ExitShort at LowestC + Average( Range, 4 ) / 2 Stop;
End
Else LowestC = 999999;
```

Inputs

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

Inputs	Default	Description
Price	Close	Price used to calculate exponential average.
Length	10	Period, defined in number of bars, to use in calculating the exponential average.
BodyLen	10	Number of bars to use in calculating the average size of the real body.

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

```
Vars: BearEngulfing(False), BullEngulfing(False), LongEntryPrice(0), ShortEntryPrice(0),
UpTrend(False), DownTrend(False), AvgBody(0), HighestC(0), LowestC(999999);
```

First, we determine whether the market is in an uptrend or a downtrend. We do this by comparing the 10-day exponential averages of the close. We compare the calculation today to the same calculation one bar ago. If today's value is greater, we consider the market to be in an uptrend and vice versa. We also calculate the average size of the real body for the last 10 bars.

EasyLanguage provides two functions, BearishEngulfing and BullishEngulfing, which find the engulfing patterns for you.

We could have used these functions in this system but decided instead to write our own criteria.

```
UpTrend = XAverage( Price, Length ) > XAverage( Price, Length )[1] ;
DownTrend = XAverage( Price, Length ) < XAverage( Price, Length )[1] ;
AvgBody = Average( AbsValue(Close - Open), BodyLen );
```

Then, we search for a Bearish or Bullish Engulfing pattern. If a Bullish Engulfing pattern is found, the BullEngulfing variable is set to true. If a Bearish Engulfing pattern is found, the BearEngulfing variable is set to true.

To find a Bullish Engulfing patterns, we first compare the close of the current bar to the open of the current bar, and the close of the previous bar to the open of the previous bar. We also compare the close of the current bar to the open of the previous bar, and the open of the current bar to the close of the previous bar. Then we make sure the current bar is at least 25% larger than the average real body and that the previous bar is at least 25% smaller than the average. We perform the similar comparisons to find the Bearish Engulfing pattern, changing the criteria accordingly.

```
BullEngulfing = ( Close > Open AND Close[1] < Open[1] ) AND
        ( Close > Open[1] AND Open < Close[1] ) AND ( Close - Open > AvgBody * 1.25 ) AND
        ( Open[1] - Close[1] < AvgBody * .75 );

BearEngulfing = ( Open > Close AND Open[1] < Close[1] ) AND ( Open > Close[1] AND
        Close < Open[1] ) AND ( Open - Close > AvgBody * 1.25 ) AND
        ( Close[1] - Open[1] < AvgBody * .75 );
```

Finally, when the market is in a downtrend and we find a Bullish Engulfing pattern, we store the high of the current bar in the variable LongEntryPrice. Likewise, when the market is in an uptrend and we find a Bearish Engulfing pattern, we store the low of the current bar in the variable ShortEntryPrice.

```
If DownTrend and BullEngulfing then LongEntryPrice = High;
If UpTrend and BearEngulfing then ShortEntryprice = Low;
```

Long Entries

To place the long entry, we use the Most Recent Occurrence (MRO) function to make sure that at least once during the last 3 bars, the market has been in a downtrend and we found a Bullish

Engulfing pattern. We also make sure we are not currently in a long position. If these two criteria are true, we place a buy stop order at the high price we captured in the variable LongEntryPrice during our setup calculations.

```
If MRO( DownTrend and BullEngulfing, 3 , 1) > -1 AND MarketPosition <> 1 then
        Buy next bar at LongEntryPrice Stop;
```

Short Entries

The short side is similar to the long side. To place the short entry, we use the MRO function to make sure that at least once during the last 3 bars, the market has been in an uptrend and we found a Bearish Engulfing pattern. We also make sure we are not currently in a short position. If these two criteria are true, we place a sell stop order at the low price we captured in the variable ShortEntryPrice during our setup calculations.

```
If MRO( UpTrend and BearEngulfing, 3 , 1) > -1 AND MarketPosition <> -1 then
        Sell next bar at ShortEntryPrice Stop;
```

As discussed in Chapter 2 of the previous volume, the MRO function enables you to specify criteria to look for, the number of occurrences that you want to look for, and the number of bars back to search. The function then returns the bar number on which the most recent occurrence of the criteria you specified occurred. It has to have occurred the specified number of times in order to return the number of the bar for the most recent occurrence. If the criteria was not found during the number of bars you specified, or the necessary number of occurrences was not found, the function returns -1.

Exit Orders

First, we use the MarketPosition function to make sure we are in a long position. When we are, we compare the current close to the value in the variable HighestC. If the current close is higher, then we store the current close in the variable. Since we initialized the variable to zero (0), the first close will always be placed in the variable. We want to capture the highest close since we entered the position. Then, we place an exit order at the highest close minus half the 4-bar average of the range. If we are not in a long position, then we set the HighestC variable to zero.

```
If MarketPosition = 1 then Begin
        If Close > HighestC then HighestC = Close;
        Exitlong at HighestC - Average( Range, 4 ) / 2 Stop;
End
Else HighestC = 0;
```

Then, we check to see if we are in a short position. When we are, we do the opposite of what we did for the long position—we compare the close of the current bar to the value in the variable LowestC. If the current close is lower, we store the current low in the variable. We initialized the variable to 999999, so the first low will always be placed in the variable. We want to capture the lowest low since we entered the position. Then we place an exit order at the lowest close plus half the 4-bar average of the range.

```
If MarketPosition = -1 then Begin
        If Close < LowestC then LowestC = Close;
        ExitShort at LowestC + Average( Range, 4 ) / 2 Stop;
End
Else LowestC = 999999;
```

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.

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

This trading system is based on an idea that is difficult to test because the pattern occurs very infrequently, making it difficult to generate statistically sound results. When we apply this trading system to an 18-year daily chart of Intel, we only get 12 trades, of which better than 55% are profitable and the ratio of winning versus losing trades is near 1.2. Figure 7 shows the System Report for the Bullish & Bearish Engulfing Patterns system on a daily Intel chart.

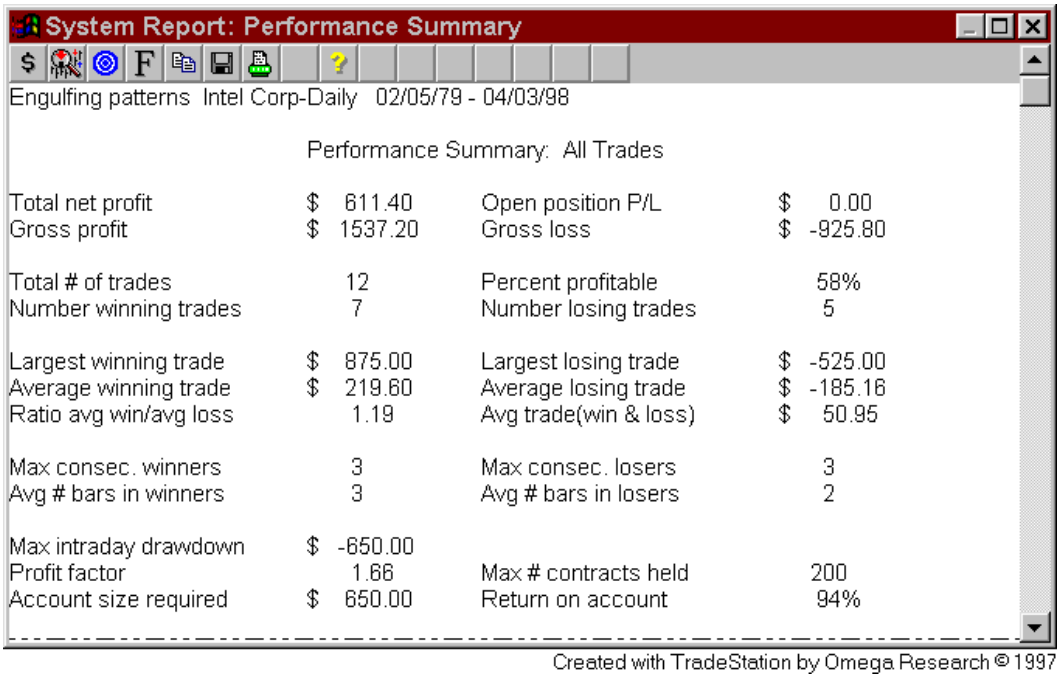


Figure 7. Sample System Report for the Bullish & Bearish Engulfing Patterns trading system

Again, we need to test this system on more data in an attempt to obtain a significant, and therefore statistically sound, number of trades.

Suggestions for Improvement

The timing of the entries seems to be acceptable; however, the exit criteria seems to be affecting the profitability of some of the trades by exiting us from the market before the big move occurs. Therefore, we may want to relax the exit criteria to give the trades some room to move. However, we have to keep in mind that this will affect our drawdown.

Also, this system attempts to pick tops and bottoms, i.e., the point at which a downtrend is reversing and vice versa, so we may want to use an additional filter like the RSI, Stochastic or another oversold/overbought oscillator to confirm the reversal.

Open-Close Histogram

The Open-Close Histogram trading system is one of the few systems we've presented that includes the opening price in its calculations. The system idea is based on the observation that in an uptrend, a market generally closes above its open, and in a downtrend, a market generally closes below its open. This relationship is certainly not true on every bar, because the market tends to go through correction periods during trending periods, but it's true on average.

Therefore, we will use the relationship between the open and close prices to determine whether the market is in an uptrend on in a downtrend. We will calculate an exponential moving average of the last 10 opens and the exponential moving average of the last 10 closes. We will then subtract the exponential average of the opens from the exponential average of the closes and draw the difference as a histogram.

When the histogram crosses from below zero to above zero, it means that the average close is greater than the average open and the market is in an uptrend; conversely, when the histogram crosses from above zero to below zero, it means that the average close is less than the average open and the market is in a downtrend.

To enter a long position, we will wait for the histogram to cross from below zero to above zero. Then, we'll place a buy stop at the high of that bar plus half the 10-bar average true range. The buy stop will remain in effect at that price for 10 bars or until the histogram crosses from above zero to below zero, whichever occurs first.

Once we enter a long position, we'll set a protective stop at the low of the bar where the histogram crossed over zero minus half of the 10-bar average true range. Also, we will exit from the long position whenever the histogram crosses under zero.

To enter a short position, we will wait for the histogram to cross from above zero to below zero. Then, we'll place a sell stop at the low of that price bar minus half of the 10-bar average true range. The sell stop will remain in effect at that price for 10-bars or until the histogram crosses from below zero to above zero, whichever occurs first.

Once we enter into a short position, we'll set our protective stop at the high of the bar on which the histogram crossed under zero plus half of the average true range. We will exit from the short position when the histogram crosses above zero.

Figure 8 shows the Open-Close Histogram system applied to a weekly chart of IBM. We also applied the OC Histogram Indicator, which we wrote and included on your CD (STAD3: OC Histogram).

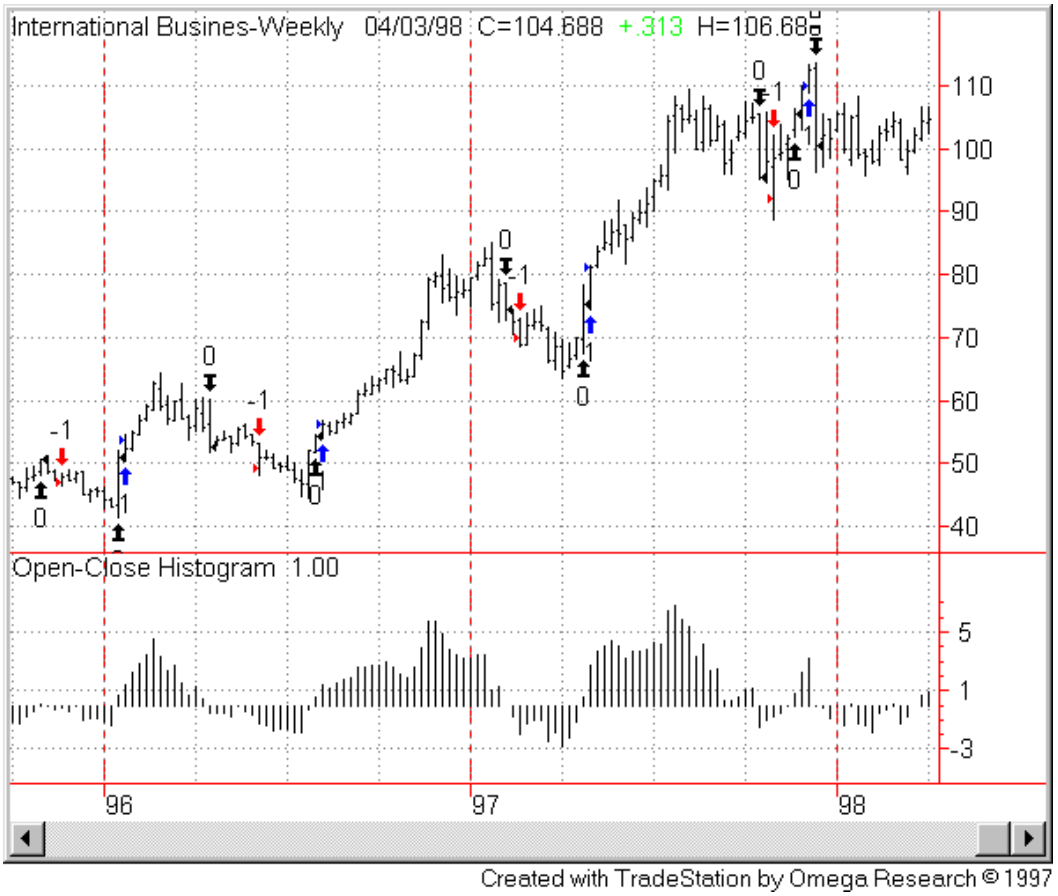


Figure 8. OC Histogram Trading System and OC Histogram Indicator applied to a weekly IBM chart

Defining your Trading Rules

In this system, we defined both long and short entries as well as exit orders. The long and short entries 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 exponential averages and calculating their difference. The setup, entry and exits are described next.

Setup

a) Calculate a 10-bar exponential average of the open prices.

b) Calculate a 10-bar exponential average of the close.

c) Subtract the average of the open prices from the average of the close.

Long Entries

a) When the histogram crosses from below zero to above zero, we will place a buy stop at the high of that bar plus half the 10-bar average true range. The buy stop will remain in effect at that price for 10 bars or until the histogram crosses from above zero to below zero, whichever occurs first.

Short Entries

a) When the histogram crosses from above zero to below zero, we'll place a sell stop at the low of that price bar minus half of the 10-bar average true range. The sell stop will remain in effect

at that price for 10-bars or until the histogram crosses from below zero to above zero, whichever occurs first.

Exits

a) We will exit from the long position whenever the histogram crosses under zero.

b) Also, on the first bar of the long position, we'll set a protective stop at the low of the bar where the histogram crossed over zero minus half of the 10-bar average true range.

c) We will exit from the short position when the histogram crosses above zero.

d) Also, once we enter into a short position, we'll set a protective stop at the high of the bar on which the histogram crossed under zero plus half of the average true range.

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: Open-Close Histogram (STAD3: OC Histogram)

```
Input: OpenLen(10), CloseLen(10);
Vars: Histogram(0), BuyPrice(0), SellPrice(0), LongExitPrice(0), ShortExitPrice(0);

{ Setup calculation - calculating the exponential averages spread }
Histogram = XAverage(Close, CloseLen) - XAverage(Open, OpenLen);

{ Long entry orders }
If Histogram Crosses Over 0 then Begin
        BuyPrice = High + Average( TrueRange, 10 ) * .5;
        LongExitPrice = Low[1] - Average(TrueRange, 10)[1]/2;
End;

If MRO(Histogram Crosses Over 0, 10, 1) > -1 AND Histogram > 0 then
        Buy ("LE") next bar at BuyPrice Stop;

{ Short entry orders }
If Histogram Crosses Under 0 then Begin
        SellPrice = Low - Average( TrueRange, 10 ) * .5;
        ShortExitPrice = High[1] + Average(TrueRange, 10)[1]/2;
End;

If MRO(Histogram Crosses Under 0, 10, 1) > -1 AND Histogram < 0 then
        Sell ("SE") next bar at SellPrice Stop;

{ Trailing stop exits }
If Histogram crosses under 0 then
        ExitLong this bar at Close;
ExitLong from entry ("LE") next bar at LongExitPrice Stop;
If Histogram Crosses Over 0 then
        ExitShort this bar at Close;
ExitShort from entry ("SE") next bar at ShortExitPrice Stop;
```

Inputs

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

Inputs	Default	Description
OpenLen	10	Period, defined in bars, of the exponential average of the open
CloseLen	10	Period, defined in bars, of the exponential average of the close

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

```
Vars: Histogram(0), BuyPrice(0), SellPrice(0), LongExitPrice(0), ShortExitPrice(0);
```

We calculate the exponential averages of the open and close prices and subtract the one from the other. We store the resulting difference in the variable called Histogram. We can then compare this variable to zero, as described later.

```
Histogram = XAverage(Close, CloseLen) - XAverage(Open, OpenLen);
```

Long Entries

We first determine whether or not the value in the variable Histogram has crossed over zero (0). This EasyLanguage instruction will look at the value of Histogram and determine if it was under zero on the previous bar and over zero on the current bar. If so, then it will set the variable BuyPrice to the high of the current bar plus half the 10-bar average of the true range. It will also set the LongExitPrice variable to the low of the previous bar minus half the 10-bar average of the true range from the previous bar.

```
If Histogram Crosses Over 0 then Begin
        BuyPrice = High + Average( TrueRange, 10 ) * .5;
        LongExitPrice = Low[1] - Average(TrueRange, 10)[1]/2;
End;
```

Then, we use the MRO function to determine if Histogram has crossed over zero once in the last 10 bars. If it has and Histogram is still greater than zero, then we place the buy stop at the value in BuyPrice.

```
If MRO(Histogram Crosses Over 0, 10, 1) > -1 AND Histogram > 0 then
        Buy ("LE") next bar at BuyPrice Stop;
```

Short Entries

We first determine whether or not the value in the variable Histogram has crossed over zero (0). This EasyLanguage instruction will look at the value of Histogram and determine if it was under zero on the previous bar and over zero on the current bar. If so, then it will set the variable BuyPrice to the high of the current bar plus half the 10-bar average of the true range. It will also set the LongExitPrice variable to the low of the previous bar minus half the 10-bar average of the true range from the previous bar.

```
If Histogram Crosses Under 0 then Begin
        SellPrice = Low - Average( TrueRange, 10 ) * .5;
        ShortExitPrice = High[1] + Average(TrueRange, 10)[1]/2;
End;
```

Then, we use the MRO function to determine if Histogram has crossed under zero once in the last 10 bars. If it has and Histogram is still less than zero, then we place the sell stop at the value in SellPrice.

```
If MRO(Histogram Crosses Under 0, 10, 1) > -1 AND Histogram < 0 then
        Sell ("SE") next bar at SellPrice Stop;
```

Exit Orders

We designed two ways of exiting our long positions. First, we will exit from our long position at the close if Histogram crosses under zero. Second, we will place a stop order at LongExitPrice.

```
If Histogram Crosses Under 0 then
        ExitLong this bar at Close;
ExitLong from Entry ("LE") next bar at LongExitPrice Stop;
```

Likewise, we designed two ways of exiting from our short positions. First, we will exit from our short positions at the close if Histogram crosses over zero. Second, we will place a stop order at ShortExitPrice.

```
If Histogram Crosses Over 0 then
        ExitShort this bar at Close;
ExitShort from Entry ("SE") next bar at ShortExitPrice Stop;
```

Notice that for the stop orders, we use the extra words from Entry ("LE") and from Entry ("SE"). Once you apply the system to your chart, take a look at the Trade-by-Trade view of the System Report. You'll notice that some exits list the letters LE or SE; these exits were generated by the stop orders. Including this text in our statements enables us to see under which circumstances the exits were made.

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.

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 exit. 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

As you can see in Figure 9, this system has posted very normal results for a trending system in a trending market. You can see the large ratio of winning versus losing trades while the percent of winning trades is near 40%. Over a period of nearly 18 years it has taken nearly 70 points.

System Report: Performance Summary

Open-Close Histogram International Busines-Weekly 01/09/70 - 04/03/98

Performance Summary: All Trades

Total net profit	$ 70.97	Open position P/L	$ -1.31
Gross profit	$ 228.55	Gross loss	$ -157.58
Total # of trades	127	Percent profitable	38%
Number winning trades	48	Number losing trades	79
Largest winning trade	$ 24.13	Largest losing trade	$ -13.56
Average winning trade	$ 4.76	Average losing trade	$ -1.99
Ratio avg win/avg loss	2.39	Avg trade(win & loss)	$ 0.56
Max consec. winners	4	Max consec. losers	5
Avg # bars in winners	14	Avg # bars in losers	4
Max intraday drawdown	$ -27.19		
Profit factor	1.45	Max # contracts held	1
Account size required	$ 27.19	Return on account	261%

Created with TradeStation by Omega Research © 1997

Figure 9. System Report for the Open-Close Histogram trading system applied to a weekly IBM chart

These results were obtained without using a stop loss (i.e., money management stop). A basic component of any trading methodology is the maximum loss you are willing to endure, a pre-defined point at which you decide that the trade is no good and that you want out of the position.

One criteria for selecting a money management stop loss is that it should not be so close that it affects a significant number of winning trades. For example, if the average winning trade has a drawdown of 2 points, and you place a stop loss order 2 points under your entry, you'll affect your winning trades—the average trade will be stopped out before it can make a profit!

You can visualize this concept very easily with the Maximum Adverse Excursion chart (available through Portfolio Maximizer, the portfolio analysis product available from Omega Research) shown in Figure 10. The chart shows all the trades, with drawdown per trade as the x-axis and profit in dollars as the y-axis.

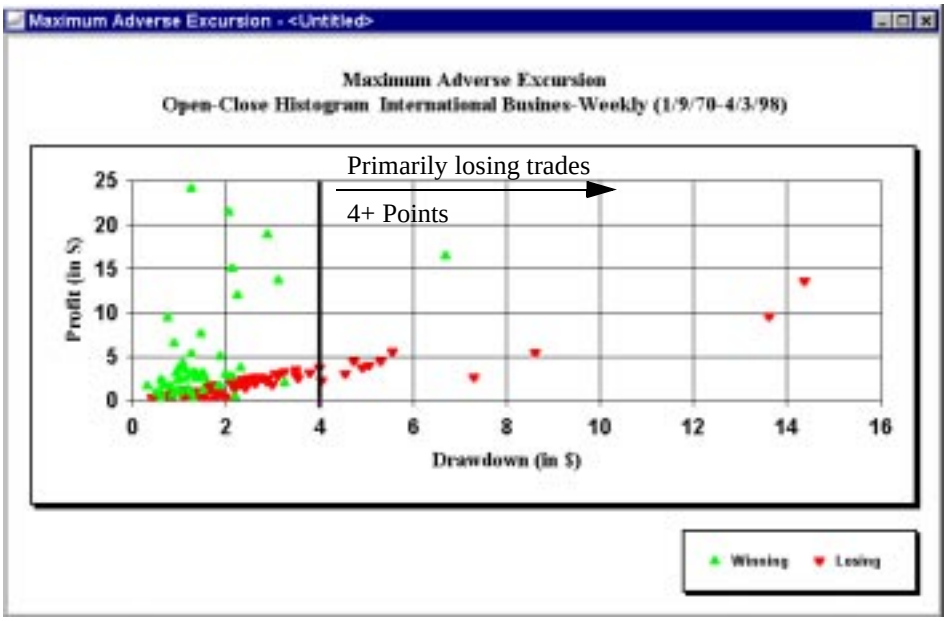


Figure 10. Maximum Adverse Excursion chart for the Open-Close Histogram system

This chart shows all trades in the test. The winning trades are represented by the up arrows, and the losing trades are represented by the down arrows. The further to the right a trade is, the greater the drawdown suffered on that trade.

You will notice that the majority of the trades to the right of 4 drawdown points per trade are losers, only one winning trade had a drawdown over 4 points versus better than 10 losing trades. It would be safe to assume that placing a stop loss at 4 points would significantly reduce your drawdown and your losses but not interfere with the winning trades.

Therefore, we decided to re-test the system this time using a 4-point money management stop. Figure 11 shows the resulting System Report. Notice that the results improved significantly; drawdown was reduced almost by half and profit improved (by reducing gross loss).

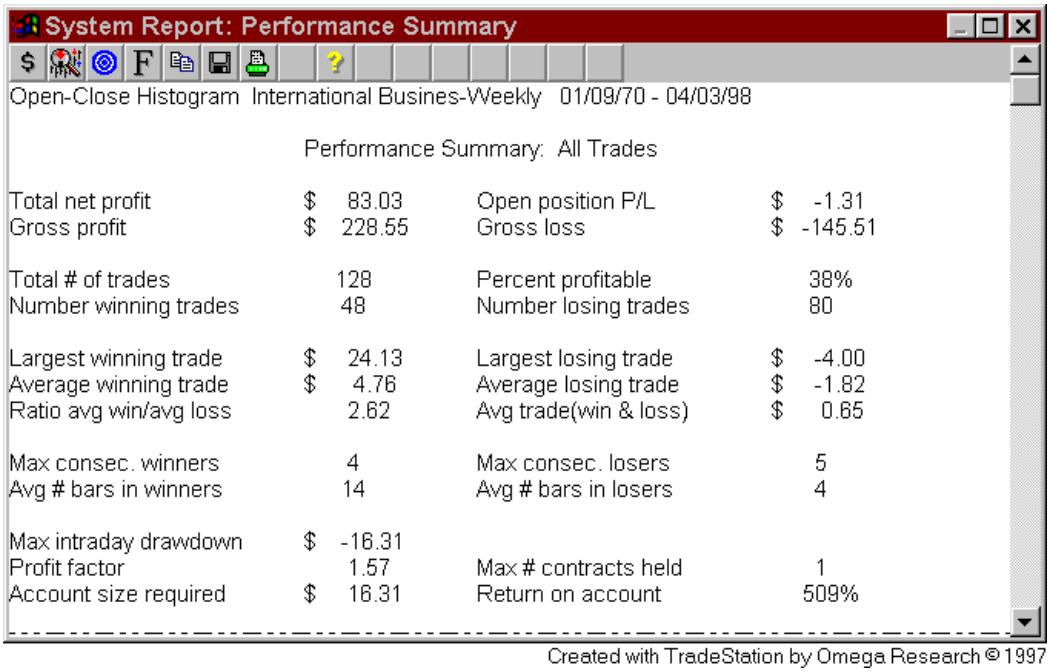


Figure 11. System Report for the Open-Close Histogram system with a 4- point money management stop

Another benefit of placing a money management stop in the correct place is that our equity curve is smoothed out significantly—we will not have as many dips as we had before, and our overall drawdown is greatly reduced.

Suggestions for Improvement

As you can see in Figure 8, this system is noticeably affected by a choppy period during the last few months of the chart. Therefore, we feel that the system can be improved by using an additional filter, such as the ADX or DMI, to verify the beginning or existence of a trend before entering trades.

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 uptrend or buying a market that is in a downtrend. 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 four Support & Resistance systems that differ in their approach but that are all designed to make the most of the sideways movement of a market.

In This Chapter

Bollinger Bands

If any assumption can be made about sideways markets, it is that over periods of time where support and resistance levels are constant, the distribution of the prices traded will tend to be normal (i.e., distributed normally in the shape of a bell curve). In other words, if we counted the instrument traded at each price, we would end up with a graph similar to the one shown in Figure 1.

Figure 1. Normal price distribution

In the above graph, the x-axis is the price, the y-axis is the frequency, and the vertical lines at the right and left are the support and resistance levels. While these support and resistance levels are not broken, prices will sometimes touch these levels and then return to the 'normal' prices. This distribution curve can be mathematically described by two numbers, the average price traded (which will be at the center of the distribution) and the standard deviation lines (which represent the support and resistance levels).

The standard deviation will indicate how much prices have varied from the average. The bigger the standard distribution number, the flatter the distribution curve, and the smaller the standard distribution, the taller the curve. Approximately 96% of the prices will fall between the two standard deviations plus or minus the average price.

If there are true support and resistance lines, we can attempt to find them by calculating Bollinger Bands, which are two standard deviations above and below the average price of the instrument analyzed. In this system, we will attempt to buy whenever the price has closed 2 bars in a row under the bottom standard deviation and sell short whenever the price closes over the top standard deviation for 2 consecutive bars.

We will allow this system to pyramid into positions, and we will buy with a limit order at the lower band and sell short with a limit order at the higher band.

Once we determined how we were going to enter the market, we turned our attention to the method of exiting the market. 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. 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 the 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.

Figure 2 shows the Bollinger Bands system applied to a chart. We also applied the standard Bollinger Bands Indicator to the chart along with the system.

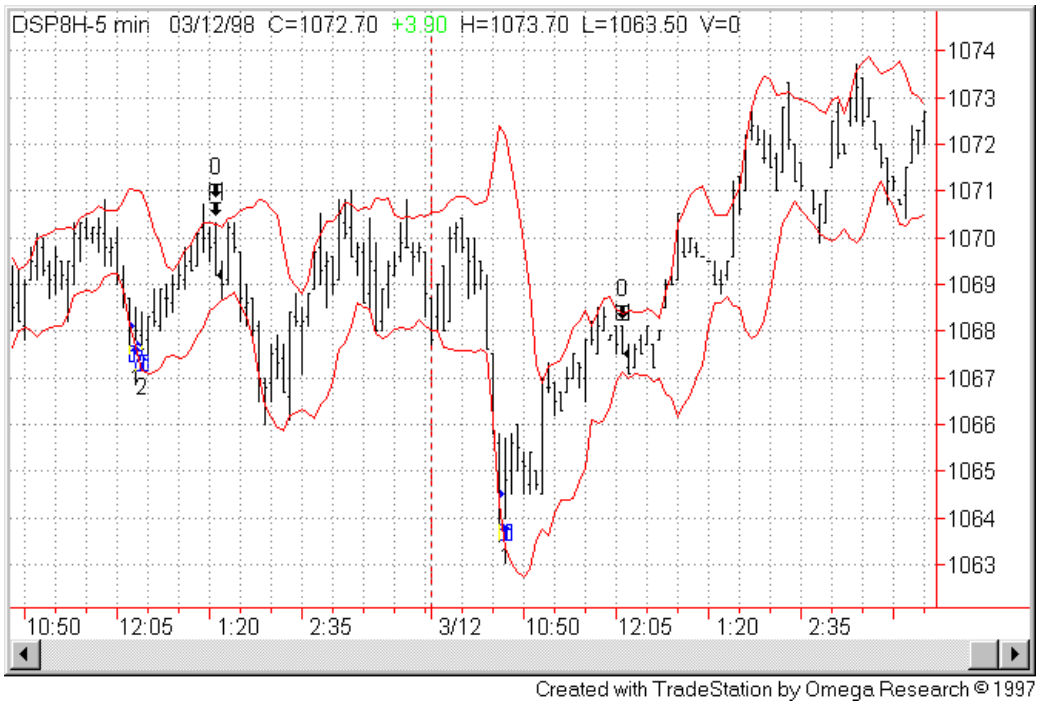


Figure 2. Bollinger Bands trading system applied to a 5-minute chart of the S&P

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 upper and lower bands. The setup, entry and exits are described next.

Setup

a) Calculate the upper and lower bands using the closing prices, a 10-bar simple moving average and 2 as the standard deviation.

b) Calculate the channel by subracting the lower band from the upper band.

Long Entries

A Buy Limit order is an order to buy at a specific price or lower.

a) Find conditions where the close has been below the lower band for the last two bars. When it has, buy at the open.

A Sell Limit order is an order to sell at a specific price or higher.

b) Once we're in a long position, place a limit order at the lower band.

Short Entries

a) Find conditions where the close has been above the upper band for two bars in a row. When it has, sell at the open.

b) Once we're in a short position, place a limit order at the upper band.

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 3 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 3 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: Bollinger Bands (STAD3: Bollinger Bds)

```
Input: NumBOvr(2), NumBUnd(2);
Vars: HiBand(0), LoBand(0), Channel(0), MP(0), StopPrice(0);

{ Setup calculations }
HiBand = BollingerBand(Close, 10, 2);
LoBand = BollingerBand(Close, 10, -2);
Channel = HiBand - LoBand;

{ Short entries }
If MRO( Close <= HiBand , NumBOvr , 1 ) = -1 AND MarketPosition <> -1 then
        Sell next bar at Open;
If MarketPosition = -1 then
        Sell ("SLim") at HiBand Limit;

{ Long entries }
If MRO( Close >= LoBand, NumBUnd , 1 ) = -1 AND MarketPosition <> 1 then
        Buy next bar at Open;
If MarketPosition = 1 then
        Buy ("BLim") at LoBand Limit;

{ Long and short trailing stops }
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 next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

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

Inputs

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

Inputs	Default	Description
NumBOvr	2	Number of consecutive bars during which the close must be over the upper band in order to generate a sell order.
NumBUnd	2	Number of consecutive bars during which the close must be under the lower band in order to generate a buy order.

Note: *When you open the system in the PowerEditor or apply it to a chart, you will see an additional input named BandLen. We don't use the input in the system. It was inadvertently left in the instructions. You can either ignore it or delete is using the PowerEditor.*

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

```
Vars: HiBand(0), LoBand(0), Channel(0), MP(0), StopPrice(0);
```

We begin by calculating the Bollinger Bands and the channel. We use the BollingerBand function and store the upper band in the variable HiBand and the lower band in the variable LoBand. We use the closing price, a 10-bar simple moving average, and a standard deviation of 2 and -2. Then we subtract the lower band from the higher band and store the resulting value in the variable Channel.

```
HiBand = BollingerBand(Close,10,2);
LoBand = BollingerBand(Close,10,-2);
Channel = HiBand - LoBand;
```

Long Entries

For our long entries, we use the MRO function to determine whether or not the close has been greater than or equal to the lower band in either of the last 2 bars. If it has not, the function will return -1, which is what we want because we are actually checking to make sure that the close has been under the lower band for the last 2 bars. We also make sure we are not currently in a long position. When these two conditions are true, we place a buy order at the open.

```
If MRO( Close >= LoBand, NumBUnd , 1 ) = -1 AND MarketPosition <> 1 then
        Buy next bar at Open;
```

Also, as soon as we are in a long position, we place a limit order at the lower band. This means that any time the price is at the lower band or under we will buy.

```
If MarketPosition = 1 then
        Buy ("BLim") at LoBand Limit;
```

Short Entries

Like we do for our long entries, we use the MRO function to determine whether or not the close has been less than or equal to the upper band in either of the last 2 bars. If it has not, the function will return -1, which is what we want because we are actually checking to make sure that the close has been over the upper band for the last 2 bars. We also make sure we are not currently in a short position. When these two conditions are true, we place a sell order at the open.

```
If MRO( Close <= HiBand , NumBOvr , 1 ) = -1 AND MarketPosition <> -1 then
        Sell next bar at Open;
```

Also, as soon as we are in a short position, we place a limit order at the upper band. This means that any time the price is at the upper band or higher we will sell short.

```
If MarketPosition = -1 then
        Sell ("SLim") at HiBand Limit;
```

Exit Orders

First we obtain our market position using the MarketPosition function and store it in the variable MP. Then, 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.

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

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

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

Then, once we are in a long position, we will place an order to exit a long position 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 plus a third of the difference between the low and the current stop price.

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

Likewise, once we are in a short position, we will 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 minus a third of the difference between the current stop price and the high of the current bar.

```
If MP = -1 then Begin
        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 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 did not enable a money management stop (the **Money Mngmnt** check 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 **Allow multiple entries in same direction from same or different signals** option. This option allows us to enter the market as many times as the signal is generated.

Testing & Improving

On a 5-minute chart of the S&P, this system behaves very much like the typical Support and Resistance system. Figure 3 shows the sample System Report.

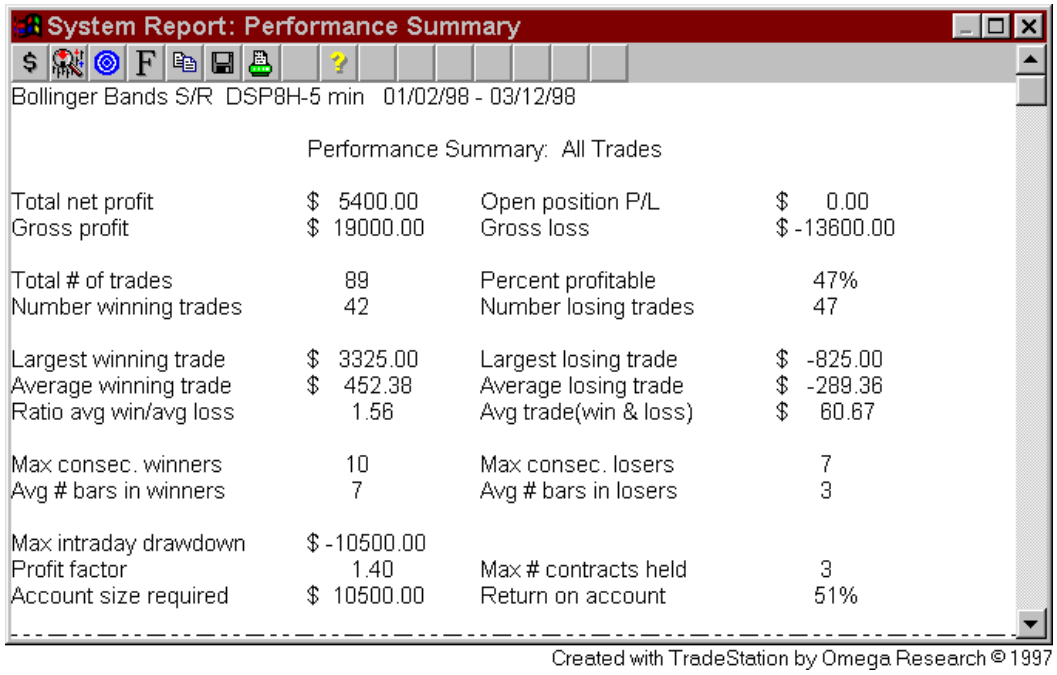


Figure 3. System Report for the Bollinger Bands system

It has almost 50% accuracy and the winning versus losing trades are near 1.5. The problem with this system is the drawdown; it requires over $10,000 in drawdown over a period of 2 months in order to return $5,400. These numbers are far from desirable.

However, keep in mind that we did not enable a money management stop. In order to determine where we should place our stop, we used the Maximum Adverse Excursion chart available in Portfolio Maximizer. Figure 4 shows the Maximum Adverse Excursion chart for the Bollinger Bands system.

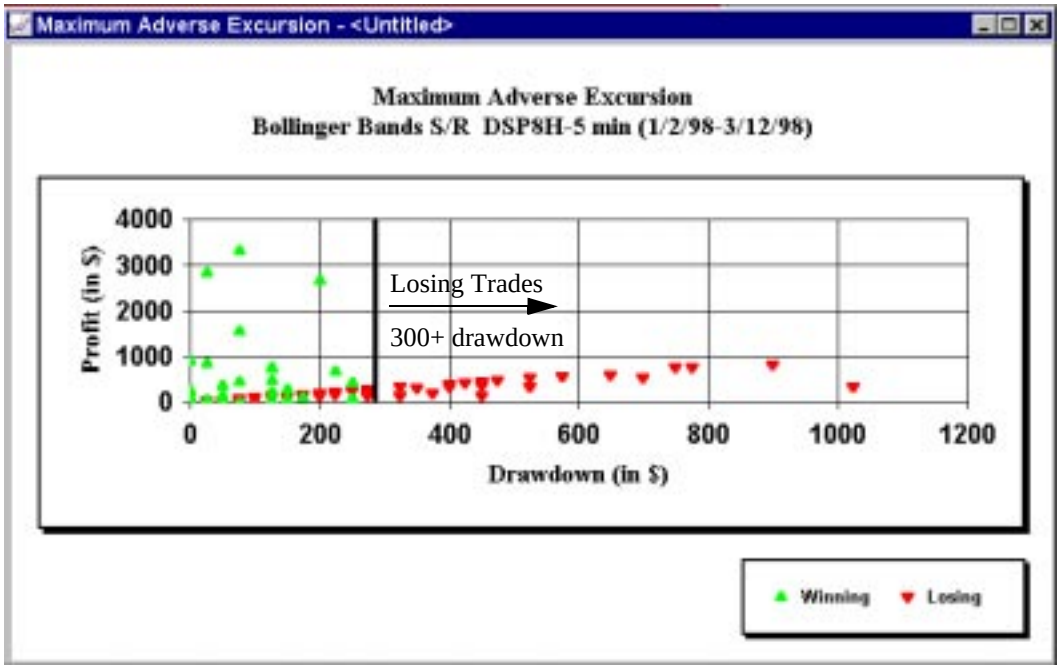


Figure 4. Maximum Adverse Excursion chart for the Bollinger Bands S/R system

By looking at the Maximum Adverse Excursion chart, we can see the maximum drawdown of every trade. We can easily identify that all trades with more than $300 drawdown have also reported losses, so $300 is the ideal number at which to place our money management stop.

Once we placed this stop, the system result improved dramatically. Figure 5 shows the new System Report.

System Report: Performance Summary

Bollinger Bands S/R DSP8H-5 min 01/02/98 - 03/12/98

Performance Summary: All Trades

Total net profit	$ 9275.00	Open position P/L	$ 0.00
Gross profit	$ 18925.00	Gross loss	$ -9650.00
Total # of trades	83	Percent profitable	47%
Number winning trades	39	Number losing trades	44
Largest winning trade	$ 3325.00	Largest losing trade	$ -300.00
Average winning trade	$ 485.26	Average losing trade	$ -219.32
Ratio avg win/avg loss	2.21	Avg trade(win & loss)	$ 111.75
Max consec. winners	10	Max consec. losers	7
Avg # bars in winners	8	Avg # bars in losers	2
Max intraday drawdown	$ -6700.00		
Profit factor	1.96	Max # contracts held	3
Account size required	$ 6700.00	Return on account	138%

Created with TradeStation by Omega Research © 1997

Figure 5. System Report for the Bollinger Bands system

We have not only reduced the drawdown but improved our profit by reducing some of the losing trades.

Suggestions for Improvement

As we have discussed in previous volumes, trending markets don't have support or resistance levels capable of turning the direction of the instrument traded, and trends are the worst enemy of Support & Resistance systems. A Support & Resistance system will repeatedly try to buy in a downtrend and sell short in an uptrend.

A way of trying to improve this and other Support & Resistance systems is to filter out trades when a strong trend is detected and relaxing the entry criteria when support or resistance levels begin to show.

On Balance True Range

The On Balance True Range system builds on the long side of the above Bollinger Bands system. One way to improve the Bollinger Bands system is to add further criteria to determine whether or not a market is oversold before taking a position. The On Balance True Range system uses the long side of the Bollinger Bands system and demonstrates the use of on balance true range as a filter.

On Balance True Range is a running total of the true range values. It is just like On Balance Volume except it is the true range that you are adding and subtracting.

When the close of the previous bar is higher than the close of the current bar, the true range of the current bar is added to the running total; when the close is less, the true range is subtracted.

As discussed in the description of the Bollinger Bands system, when the market becomes extremely oversold, prices tend to get close to the lower band, and as soon as two consecutive closes are below the lower band, we establish a long position. For this system, however, we will add the condition that the on balance true range must also be under its 9-bar exponential average.

Our reasoning is that since we are looking for the market to be oversold in order to try to buy at the bottom, the on balance true range line will decline more rapidly as the market closes down in bars that have larger ranges, thus providing an even stronger indication that the market is oversold.

We'll use the same exit technique as the Bollinger Bands 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.

Defining your Trading Rules

In this system, we defined long entries only as well as exit orders. We also performed some setup work, which involved calculating the upper and lower bands as well as the on balance true range. The setup, entry and exits are described next.

Setup

a) Calculate the on balance true range for the current bar and compare it to the 9-bar exponential average of the on balance true range. If it is less, continue with the remaining instructions, if not, then don't.

b) Calculate the upper and lower bands using the closing prices, a 10-bar simple moving average, and 2 as the standard deviation.

c) Calculate the channel by subracting the lower band from the upper band.

Long Entries

a) Find conditions where the close has been below the lower band for the last two bars. When it has, buy at the open.

b) Once we're in a long position, place a limit order at the lower band.

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.

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 True Range (STAD3: OnBal True Rg)

```
Input: NumBOvr(2), NumBUnd(2);
Vars: HiBand(0), LoBand(0), Channel(0), MP(0), StopPrice(0);

If OBTR < XAverage(OBTR, 9) then Begin
        HiBand = BollingerBand(Close,10,2);
        LoBand = BollingerBand(Close,10,-2);
        Channel = HiBand - LoBand;
        { Long Entries }
        If MRO( Close >= LoBand, NumBUnd , 1 ) = -1 AND MarketPosition <> 1 then
                Buy next bar at Open;
        If MarketPosition = 1 then
                Buy ("BLim") at LoBand Limit;
        { Long and Short Trailing Stop }
        MP = MarketPosition;
        If MP = 1 and MP[1] <> 1 then
                StopPrice = Low - Average(Range,4);
        If MP = 1 then begin
                Exitlong next bar at StopPrice Stop;
                StopPrice = StopPrice + (Low - StopPrice)/3;
        End;
End;
```

Inputs

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

Inputs	Default	Description
NumBOvr	2	Number of consecutive bars during which the close must be over the upper band in order to generate a sell order.
NumBUnd	2	Number of consecutive bars during which the close must be under the lower band in order to generate a buy order.

Note: *When you open the system in the PowerEditor or apply it to a chart, you will see an additional input named BandLen. We don't use the input in the system. It was inadvertently left in the instructions. You can either ignore it or delete is using the PowerEditor.*

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

```
Vars: HiBand(0), LoBand(0), Channel(0), MP(0), StopPrice(0);
```

We begin by calculating the on balance true range for the current bar and comparing it to the 9-bar exponential average of the on balance true range. We use the OBTR function. If the current OBTR is less than the 9-bar exponential average, then we will proceed. We only evaluate the remaining instructions when the condition is met.

When the on balance true range is less than the 9-bar exponential average, we then calculate the Bollinger Bands and the channel. We use the BollingerBand function and store the upper band in the variable HiBand and the lower band in the variable LoBand. We use the closing price, a 10-bar simple moving average, and a standard deviation of 2 and -2. Then we subtract the lower band from the higher band and store the resulting value in the variable Channel.

```
If OBTR < XAverage(OBTR, 9) then Begin
        HiBand = BollingerBand(Close,10,2);
        LoBand = BollingerBand(Close,10,-2);
        Channel = HiBand - LoBand;
```

Long Entries

For our long entries, we use the MRO function to determine whether or not the close has been greater than or equal to the lower band in either of the last 2 bars. If it has not, the function will return -1, which is what we want because we are actually checking to make sure that the close has been under the lower band for the last 2 bars. We also make sure we are not currently in a long position. When these two conditions are true, we place a buy order at the open.

```
        If MRO( Close >= LoBand, NumBUnd, 1 ) = -1 AND MarketPosition <> 1 then
                Buy next bar at Open;
```

Also, as soon as we are in a long position, we place a limit order at the lower band.

```
        If MarketPosition = 1 then
                Buy ("BLim") at LoBand Limit;
```

Exit Orders

First we obtain our market position using the MarketPosition function and store it in the variable MP. Then, 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.

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

Then, once we are in a long position, we will place an order to exit a long position 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 plus a third of the difference between the low and the current stop price.

```
        If MP = 1 then Begin
                Exitlong 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 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.

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 **Allow multiple entries in same direction by same or different signals** option. This option allows us to enter the market as many times as the signal is generated.

Testing & Improving

Checking the on balance true range against its exponential moving average can serve as a very effective filter for other signals. Figure 6 shows the System Results for the long side of the system provided as Bollinger Bands during a period of 2 months on a 5-minute chart of the S&P.

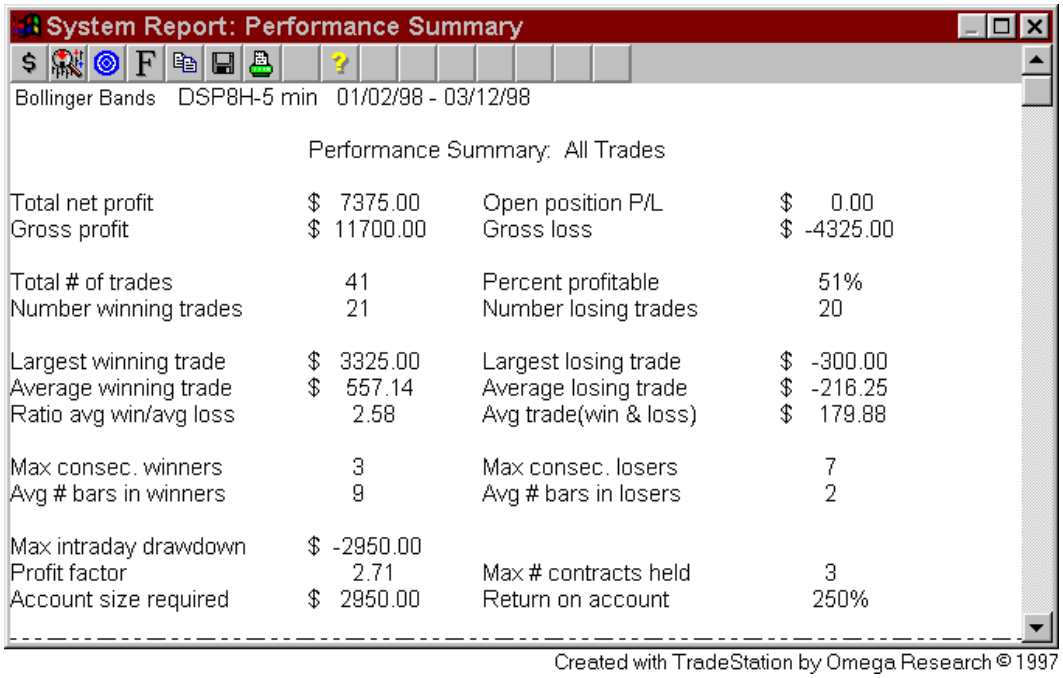
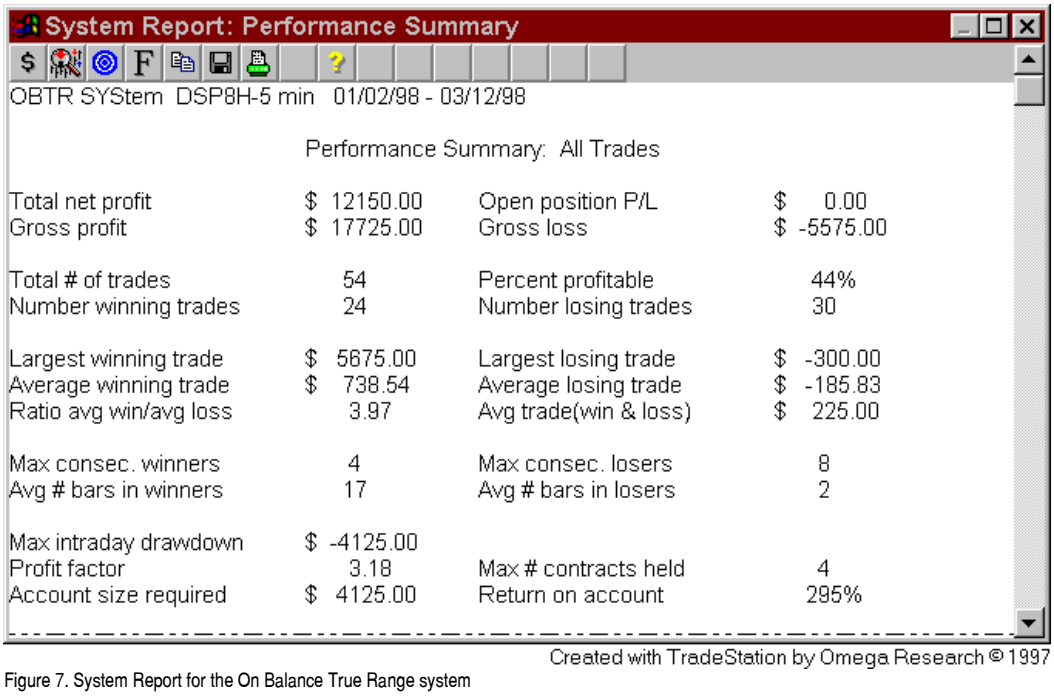


Figure 6. System Report for the long side of the Bollinger Bands system

The results are quite acceptable, with 51% winning trades and almost a 2.5 ratio of average winning versus losing trade. However, once we add a condition to only take trades when the on balance true range is under its 9-bar exponential average, we obtain substantially better results, with better profits, improved numbers in winning versus losing trades and a lower average losing trade. Figure 7 shows the System Report for the On Balance True Range system.


System Report: Performance Summary

OBTR SYStem DSP8H-5 min 01/02/98 - 03/12/98

Performance Summary: All Trades

Total net profit	$ 12150.00	Open position P/L	$ 0.00
Gross profit	$ 17725.00	Gross loss	$ -5575.00
Total # of trades	54	Percent profitable	44%
Number winning trades	24	Number losing trades	30
Largest winning trade	$ 5675.00	Largest losing trade	$ -300.00
Average winning trade	$ 738.54	Average losing trade	$ -185.83
Ratio avg win/avg loss	3.97	Avg trade(win & loss)	$ 225.00
Max consec. winners	4	Max consec. losers	8
Avg # bars in winners	17	Avg # bars in losers	2
Max intraday drawdown	$ -4125.00		
Profit factor	3.18	Max # contracts held	4
Account size required	$ 4125.00	Return on account	295%

Created with TradeStation by Omega Research © 1997

Figure 7. System Report for the On Balance True Range system

Suggestions for Improvement

Again, any time you are working with a Support & Resistance system, it is essential that you identify a trending period and attempt to limit losses in this type of market. One way this can be done is by filtering out trades when in a strong trend is detected and relaxing the entry criteria when support or resistance levels begin to show.

Micro-M & Micro-W Formations

Micro-M Tops and Micro-W Bottoms graphically depict the battle between the bulls and bears at market turning points. When used with oscillators (e.g., Accumulation Distribution or RSI), Micro-Ms and Micro-Ws help find significant turning points in the market.

For a Micro-W Bottom, we first identify a bullish divergence low (a new low in price accompanied by a higher low in an oscillator). Once we find the divergence, we look for a bar that closes above the previous day's close, followed immediately by a bar that closes lower. After the up-day/down-day sequence, we will buy when prices rally above the high of the up/down pattern. In other words, we want to buy at the higher of the 2 days in the up/down sequence.

For a Micro-M top, we first identify a bearish divergence high (a new high in price that is accompanied by a lower high in an oscillator). Once the divergence is found, we look for a bar that closes below the previous day's close, followed immediately by a bar that closes higher. After the down-day/up-day sequence, we will sell short when prices fall bellow the low of the down/up pattern. In other words, we want to sell short at the lower of the 2 days in the down-bar/up-bar sequence. Figure 8 illustrates this graphically.

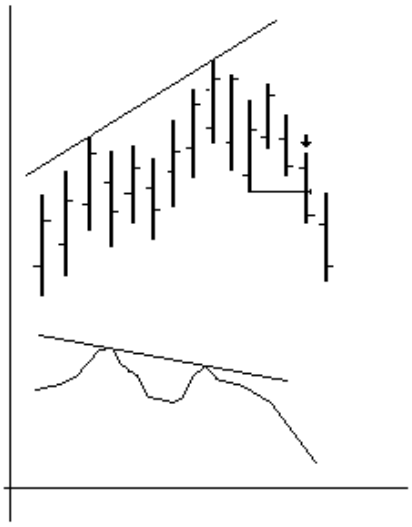


Figure 8. Bearish divergence high found followed by a down-day/up-day sequence

Once we determined how we were going to enter the market, we turned our attention to the method of exiting the market. We will use trailing stops to exit from the positions. 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. 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

We wrote this system in three parts. We wrote one system for the long side, called Micro W Bot, and one system for the short side, called Micro M Top. The third system, called Micro-M & Micro-W Formations (STAD3: MM & MW), combines these two together into one using the Include System statement.

The long and short sides are virtually identical except that they are reversed. Therefore, while we explain the setup, entries and exits for the systems when combined as well as show the EasyLanguage instructions for all three systems, we will only break down the short side, or the STAD3: Micro M Top system.

Setup

a) First we find and store the last 10 swing highs in the price data and do the same for the oscillator. We also find and store the last 10 swing lows in the price data and do the same for the oscillator.

b) We also find the mates for the swing highs and lows. To be considered mates, the swings much be 5 bars or less away from each other.

b) Once the mates are identified, we look for a divergence between the price and the oscillator. We will look for both bullish low and bearish high divergences. A bullish divergence low looks at the swing lows and is defined as a new low in price accompanied by a higher low in an

oscillator, and a bearish divergence high looks at the swing highs and is defined as a new high in price that is accompanied by a lower high in an oscillator.

Long Entries

a) Once we find a bullish divergence low, we look for a bar that closes above the previous day's close, followed immediately by a bar that closes lower. After the up-day/down-day sequence, we will buy when prices fall below the low of the up/down pattern.

Short Entries

a) Once we find a bearish divergence high, we look for a bar that closes below the previous day's close, followed immediately by a bar that closes higher. After the down-day/up-day sequence, we will sell short when prices fall bellow the low of the down/up pattern.

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.

Designing & Formatting

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

EasyLanguage Instructions: Micro-M & Micro-W Formations (STAD3: MM & MW)

```
Input: Price(AccumDist(Volume)), StrPrc(4), StrPrc2(4);

IncludeSystem:"STAD3: Micro M Top", Price, StrPrc, StrPrc2;
IncludeSystem:"STAD3: Micro W Bot", Price, StrPrc, StrPrc2;
```

EasyLanguage Instructions: Micro W Bottoms (STAD3: Micro W Bot)

```
Input: Price(AccumDist(V)), StrPrc(4), StrPrc2(4);
Array: SL1[10,4](-999), SL2[10,3](-999);
Vars: Count(0), Count2(0), NewSwing(False), Found(False), Done(False), SW1(0), SW2(0),
BarsSincePattern(0), MP(0), StopPrice(0);

{ Following new swing lows in price }
NewSwing = False;
If SwingLowBar(1,Low,StrPrc,StrPrc+2) = StrPrc then Begin
        For Count = 10 DownTo 1 Begin
                SL1[Count,0] = SL1[Count-1,0];
                SL1[Count,1] = SL1[Count-1,1];
```

```
                SL1[Count,2] = SL1[Count-1,2];
                SL1[Count,3] = SL1[Count-1,3];
                SL1[Count,4] = SL1[Count-1,4];
        End;
        SL1[0,0] = Low[StrPrc];
        SL1[0,1] = BarNumber - StrPrc;
        SL1[0,2] = Date[StrPrc];
        SL1[0,3] = Time[StrPrc];
        SL1[0,4] = 0;
        NewSwing = True;
End;

{ Following new swing lows in the oscillator }
If SwingLowBar(1,Price,StrPrc2,StrPrc2+2)  = StrPrc2 then Begin
        For Count = 10 DownTo 1 Begin
                SL2[Count,0] = SL2[Count-1,0];
                SL2[Count,1] = SL2[Count-1,1];
                SL2[Count,2] = SL2[Count-1,2];
                SL2[Count,3] = SL2[Count-1,3];
        End;
        SL2[0,0] = Price[StrPrc2];
        SL2[0,1] = BarNumber - StrPrc2;
        SL2[0,2] = Date[StrPrc2];
        SL2[0,3] = Time[StrPrc2];
        NewSwing = True;
End;

{ Determining divergence in swings }
Found = False;
If NewSwing then Begin
        If AbsValue( SL1[0,1] - SL2[0,1] ) <= 5 then Begin
                For Count = 1 to 10 Begin
                        For Count2 = 1 to 10 Begin
                                If AbsValue( SL1[Count,1] - SL2[Count2,1] ) <= 5 AND
                                SL1[0,4] = 0 AND( ( SL1[0,0] > SL1[Count,0] AND
                                SL2[0,0] < SL2[Count2,0] ) OR ( SL1[0,0] < SL1[Count,0] AND
                                SL2[0,0] > SL2[Count2,0] ) ) then Begin
                                        SW1 = Count;
                                        SW2 = Count2;
                                        Found = True;
                                        Count2 = 11;
                                        Count = 11;
                                        SL1[0,4] = -1;
                                End;
                        End;
                End;
        End;
End;

{ Drawing trendline on the price on the divergence points - commented out }
{
If Found then Begin
        Value1 = TL_New(SL1[SW1,2] , SL1[SW1,3], SL1[SW1,0], SL1[0,2], SL1[0,3], SL1[0,0] );
End;
}
```

Notice that we commented out the trendline statement. If you want the system to draw trendlines when it identifies divergences, you can remove the curly brackets.

```
{ Long entry order }
BarsSincePattern = MRO( Close[1] > Close[2] AND Close < Close[1], 3, 1);
If MRO( Found, 5, 1) > -1 AND BarsSincePattern > 0 then
        Buy next bar at Highest(High, 4) Stop;

{ Long exit orders }
MP = MarketPosition;
If MP = 1 and MP[1] <> -1 then
        StopPrice = Low - Average(Range,4);
If MP = 1 then Begin
        ExitLong next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;
```

EasyLanguage Instructions: Micro M Top (STAD3: Micro M Top)

```
Input: Price(RSI(Close,10)), StrPrc(4), StrPrc2(4);
Array: SH1[10,4](-999), SH2[10,3](-999);
Vars: Count(0), Count2(0), NewSwing(False), Found(False), Done(False), SW1(0), SW2(0),
BarsSincePattern(0), MP(0), StopPrice(0);

{ Following swing highs in price }
NewSwing = False;
If SwingHighBar(1,High,StrPrc,StrPrc+2) = StrPrc then Begin
        For Count = 10 DownTo 1 Begin
                SH1[Count,0] = SH1[Count-1,0];
                SH1[Count,1] = SH1[Count-1,1];
                SH1[Count,2] = SH1[Count-1,2];
                SH1[Count,3] = SH1[Count-1,3];
                SH1[Count,4] = SH1[Count-1,4];
        End;
        SH1[0,0] = High[StrPrc];
        SH1[0,1] = BarNumber - StrPrc;
        SH1[0,2] = Date[StrPrc];
        SH1[0,3] = Time[StrPrc];
        SH1[0,4] = 0;
        NewSwing = True;
End;

{ Following swing highs in the oscillator }
If SwingHighBar(1,Price,StrPrc2,StrPrc2+2)  = StrPrc2 then Begin
        For Count = 10 DownTo 1 Begin
                SH2[Count,0] = SH2[Count-1,0];
                SH2[Count,1] = SH2[Count-1,1];
                SH2[Count,2] = SH2[Count-1,2];
                SH2[Count,3] = SH2[Count-1,3];
        End;
        SH2[0,0] = Price[StrPrc2];
        SH2[0,1] = BarNumber - StrPrc2;
        SH2[0,2] = Date[StrPrc2];
        SH2[0,3] = Time[StrPrc2];
        NewSwing = True;
End;
```

```
{ Determining divergence between the price and the oscillator }
Found = False;
If NewSwing then Begin
        If AbsValue( SH1[0,1] - SH2[0,1] ) <= 5 then Begin
                For Count = 1 to 10 Begin
                        For Count2 = 1 to 10 Begin
                                If AbsValue( SH1[Count,1] - SH2[Count2,1] ) <= 5 AND
                                SH1[0,4] = 0 AND ( ( SH1[0,0] > SH1[Count,0] AND
                                SH2[0,0] < SH2[Count2,0] ) OR
                                ( SH1[0,0] < SH1[Count,0] AND
                                SH2[0,0] > SH2[Count2,0] ) ) then Begin
                                        SW1 = Count;
                                        SW2 = Count2;
                                        Found = True;
                                        Count2 = 11;
                                        Count = 11;
                                        SH1[0,4] = -1;
                                End;
                        End;
                End;
        End;
End;
{ Drawing trendline on the price on the divergence points - commented out }
{
If Found then Begin
        Value1 = TL_New(SH1[SW1,2] , SH1[SW1,3], SH1[SW1,0], SH1[0,2], SH1[0,3], SH1[0,0] );
End;

}
{ Short entry orders }
BarsSincePattern = MRO( Close[1] < Close[2] AND Close > Close[1], 3, 1);
If MRO( Found, 5, 1) > -1 AND BarsSincePattern > 0 then
        Sell next bar at Lowest(Low,4) Stop;

{ Trailing stops }
MP = MarketPosition;
if MP = -1 and MP[1] <> -1 then
        StopPrice = High + Average(Range,4);
If MP = -1 then Begin
        Exitshort next bar at StopPrice Stop;
        StopPrice = StopPrice - (StopPrice - High)/3;
End;
```

Inputs

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

Inputs	Default	Description
Price	RSI(Close, 20)	Oscillator or other value to compare against price.
StrPrc	4	Strength, expressed in bars, of the swing high bar. Used for the price data.
StrPrc2	4	Strength, expressed in bars, of the swing high bar. Used for the oscillator.

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

```
Array: SH1[10,4](-999), SH2[10,3](-999);
Vars: Count(0), Count2(0), NewSwing(False), Found(False), Done(False), SW1(0), SW2(0),
BarsSincePattern(0), MP(0), StopPrice(0);
```

Notice that we defined 2 arrays, SH1 and SH2. We will use these arrays to store 4 pieces of information for each of the last 10 swing highs in both the price data and the oscillator values, respectively. For each, we will store the high price, bar number, date and time of the swing high bar.

We begin by setting the variable NewSwing to false. Whenever we find a new swing, either in the price or in the oscillator, we set the variable to true. In this way, it acts as a flag. We use this flag later when we look at the divergence between the price and the oscillator.

Then we use the function SwingHighBar to search for swing high bars. Every time we find a new swing high in the prices, we discard the swing in element 10, put the swing that was in element 9 into element 10, put the swing in that was in element 8 into element 9 and so on until we move the swing in element 0 to element 1. Finally, we put the new swing into element 0. This process is called down buffering.

You'll notice that in addition to the four pieces of information, we store a fifth piece of information. We call this the flag cell. We set it to zero whenever we find a new swing high, and we set it to -1 whenever the particular swing high is used as the most recent swing for a divergence. Once we have found a divergence using that swing high as the most recent swing high, we don't want to re-use the swing high again as the most recent.

```
NewSwing = False;
If SwingHighBar(1,High,StrPrc,StrPrc+2) = StrPrc then Begin
        For Count = 10 DownTo 1 Begin
                SH1[Count,0] = SH1[Count-1,0];
                SH1[Count,1] = SH1[Count-1,1];
                SH1[Count,2] = SH1[Count-1,2];
                SH1[Count,3] = SH1[Count-1,3];
                SH1[Count,4] = SH1[Count-1,4];
        End;
        SH1[0,0] = High[StrPrc];
        SH1[0,1] = BarNumber - StrPrc;
        SH1[0,2] = Date[StrPrc];
        SH1[0,3] = Time[StrPrc];
        SH1[0,4] = 0;
        NewSwing = True;
End;
```

If a swing high bar is found, the remaining instructions are evaluated.

A For loop is used to down buffer the elements in the array, so the information for the most recent array can be stored in element 0.

The information for the swing high bar is stored in element 1 of the array. The fifth cell is set to zero, indicating that it is a new swing high.

Finally, NewSwing is set to true because a swing high was found.

We do the same thing with the oscillator. The only difference is that we don't use the flag cell. The array for the oscillator doesn't use the flag because we may want to use the same swing in the oscillator to look for divergence with more than one swing of the price.

```
If SwingHighBar(1,Price,StrPrc2,StrPrc2+2)  = StrPrc2 then Begin
        For Count = 10 DownTo 1 Begin
                SH2[Count,0] = SH2[Count-1,0];
                SH2[Count,1] = SH2[Count-1,1];
                SH2[Count,2] = SH2[Count-1,2];
                SH2[Count,3] = SH2[Count-1,3];
```

```
        End;
        SH2[0,0] = Price[StrPrc2];
        SH2[0,1] = BarNumber - StrPrc2;
        SH2[0,2] = Date[StrPrc2];
        SH2[0,3] = Time[StrPrc2];
        NewSwing = True;
End;
```

Following is a table illustrating how an array stores the information. This table corresponds to the SH1 array, which stores the swing high bar information for the price data.

Element	High Price	Bar Number	Date	Time	Flag
0	650	250	Date	Time	0
1	625	220	Date	Time	0
2	660	180	Date	Time	0
3	599	150	Date	Time	0
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
10	580	40	Date	Time	0

Next, we set the variable called Found to false and determine whether or not we have found a new swing in either the price or the oscillator by checking the variable NewSwing to determine if its value is true (Line 1 on the next page). If we have found a new swing high, we check to see if the most recent swing of the price and the most recent swing of the oscillator are at most 5 bars from each other (Line 2). If they are, we start looking for mates, meaning a swing high in price and a swing high in the oscillator that are 5 bars or less apart (Line 5), and we check to see if the most recent swing high of prices has been used (Line 6). We don't want to use a swing high in price as the most recent swing more than once. Then we check for divergence, where within the mates we found either the price is going up and the oscillator is going down (Line 7) OR the price is going down and the oscillator is going up (Line 8).

If all of these conditions are met, we will save the position of the swings that met our criteria into variables SW1 and SW2 for later use (Lines 9 and 10), we change the variable Found to true (meaning we found a divergence, Line 11), we set the Count variables to 11 to exit the loops (Line12 and 13), and we store a -1 in the flag cell of the most recent swing high of the price array (Line 14) to indicate that we have used it already in our comparison.

We can accomplish all this comparing easily using the nested loops in Lines 3 & 4. They let us traverse the arrays using the Count variable to point at the array of prices and the Count2 variable to point at the array containing the oscillator information.

The way the loops work is that we run the inner loop with Count1 = 1. With Count1 = 1 we move Count2 from 1 to 10, thereby comparing the first element in the price array with all 10 in the oscillator array. Once the inner loop completes for Count2 = 1 to 10, then the outer loop begins its second loop with Count1 = 2, and we again go through the inner loop with Count2 going from 1 to 10. Again, the effect is that we compare element number 1 of the of the price array with all 10 elements of the oscillator, then compare element number 2 of the price array with all 10 elements of the oscillator, and so on until we have compared all 10 elements of the price array to the elements in the oscillator array.

```
    { Determining divergence between the price and the oscillator }
    Found = False;
1   If NewSwing then Begin
2       If AbsValue( SH1[0,1] - SH2[0,1] ) <= 5 then Begin
3           For Count = 1 to 10 Begin
4               For Count2 = 1 to 10 Begin
5                   If AbsValue( SH1[Count,1] - SH2[Count2,1] ) <= 5 AND
6                   SH1[0,4] = 0 AND
7                   (( SH1[0,0]>SH1[Count,0] AND SH2[0,0]<SH2[Count2,0] ) OR
8                   ( SH1[0,0] < SH1[Count,0] AND SH2[0,0] > SH2[Count2,0] )) then Begin
9                       SW1 = Count;
10                      SW2 = Count2;
11                      Found = True;
12                      Count2 = 11;
13                      Count = 11;
14                      SH1[0,4] = -1;
15                  End;
16              End;
17          End;
18      End;
19  End;
```

Long Entries

We use the MRO function to determine whether or not the down-day/up-day pattern occurred wthin the last 3 bars. If it did, the MRO function will return the bar number on which the pattern was completed, and we'll store this number in the varible BarsSincePattern.

We also use the MRO function to determine whether or not the variable Found has been true once at any time during the last 5 bars (the variable Found is true once a divergence is found), and we look to make sure that the BarsSincePattern variable holds a value greater than 0.

When these conditions are true (there was a divergence followed by the down-day/up-day pattern), then will place a sell stop order at the lowest low of the last 4 bars.

```
BarsSincePattern = MRO( Close[1] < Close[2] AND Close > Close[1], 3, 1);
If MRO( Found, 5, 1) > -1 AND BarsSincePattern > 0 then
        Sell next bar at Lowest(Low,4) Stop;
```

Exit Orders

First we obtain our market position using the MarketPosition function and store it in the variable MP. Then, on 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.

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

Then, once we are in a short position, we will place an order to exit a short position 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 of the current bar.

```
If MP = -1 then Begin
        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 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

As you can see in Figure 9, the System Report shows results somewhat different from a traditional Support & Resistance system, mainly because of the high ratio of average winning trade to losing trades (3.7). Yet the largest winning trade is nearly 9 points while the average is around 3. This can be a signal that a few big trades might be influencing the system results, which can be significant in Support & Resistance or Volatility Expansion systems.

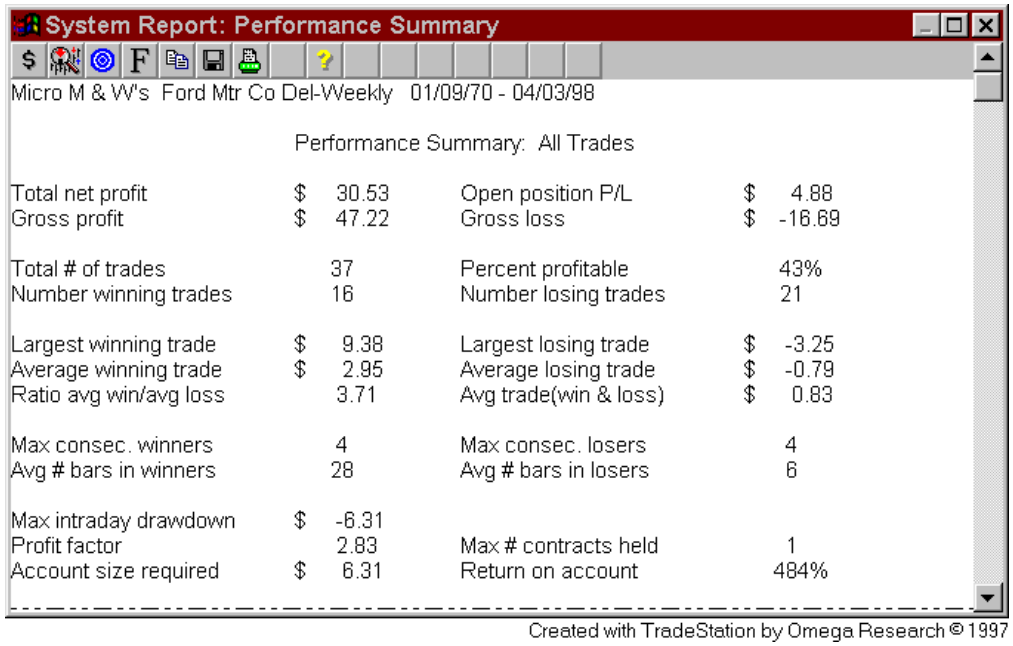


Figure 9. Sample System Report for the Micro-M & Micro-W Formations system

One of the ways of looking at this is by studying the win/loss analysis available in Portfolio Maximizer. In the Winning Trade Analysis section of the Individual Report, shown in Figure 10, you can see that the standard deviation of the winning trades was 3.16 points, even greater than the average winning trade at 2.95.

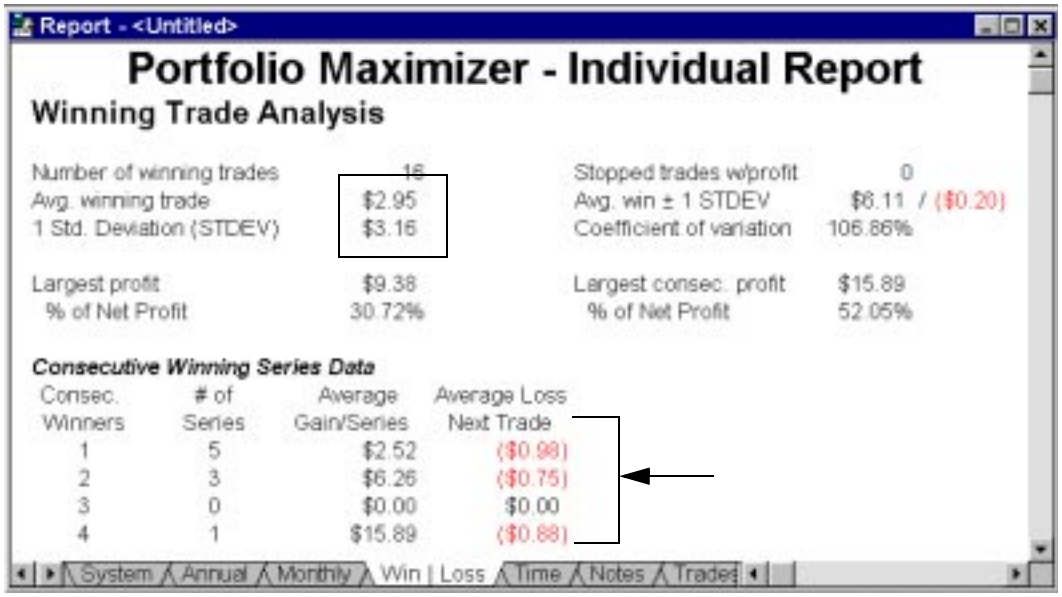


Figure 10. Winning Trade Analysis of the Micro-M & Micro-W Formations system

What this shows is some instability or inconsistency in the performance of the winning trades. As a rule of thumb, you will want to see as small a number as possible in the winning trade standard deviation (meaning that the trades have been stable and close to the average). However, this report also shows several positive numbers. If you look at the Consecutive Winning Series Data, the trades that stopped our string of winning trades were significantly smaller than the gain. This points to big rallies being interrupted by small losing trades. Also, take a look at the Losing Trade Analysis section, shown in Figure 11.

Figure 11. Losing Trade Analysis of the Micro-M & Micro-W Formations system

It shows a small average losing trade and an acceptable standard deviation for the losing trades. Also, notice how each losing streak was ended by a sizeable winner.

We had a string of four losers that gave us losses over 4 points, yet this series of bad trades was terminated by a winning trade that made 4 ¾ points. Trades like these will make the system easier to trade.

Suggestions for Improvement

The percent of winning versus losing trades seems a little low for a Support & Resistance system; it might be worth trying to use an additional filter in an attempt to identify the weakening of a strong trend. We could also try using a second filter to try an determine when the market we're trading is extremely overbought or oversold.

Variable Zones

One of the most common problems with any overbought/oversold oscillator is that it uses fixed values when determining if the oscillator is in oversold or overbought territory. This causes the oscillator, and any trading system that is derived from this oscillator, to become ineffective if the market moves away from the bands or when used on a market that oscillates between different values.

To avoid this limitation, we can use a dynamic or variable method for calculating the overbought and oversold bands. For example, we can have EasyLanguage keep track of the last 10 significant turning points of any oscillator we follow and use the average turning points as the overbought and the oversold zones instead of fixed values.

This technique can be used with any indicator, formula or even price data that has a tendency to oscillate around a central value.

The system is written with inputs, so you can change the indicator or formula on which to base the bands.

Calculating the average turning point will not necessarily be sufficient, as the resulting lines might be crossed often by the oscillator. In order to increase the distance of these lines from the normal price of the oscillator, we can also have EasyLanguage store the highest turning point of the oscillator and use an intermediate value between this highest high and the average as the overbought line. We can do the same for the oversold line.

The general principle is as follows:

```
Overbought Zone: Average Top + ( ( Highest Top - Average Top ) * Factor )
Oversold Zone:Average Bottom - ( ( Lowest Bottom - Average Bottom ) * Factor )
```

For the purposes of this system, we used the RSI Indicator and used this dynamic method of calculation to determine the line values, or zones. We will buy whenever the RSI crosses over the oversold zone and we will sell short whenever the RSI crosses under the overbought zone.

Figure 12 shows the Variable Zones system applied to a 5-minute chart of the S&P. We also applied the Variable Zones indicator we wrote to visualize the bands (STAD3: Variable Zns). Take a look at the indicator; notice how the bands adjust to the movement of the RSI.

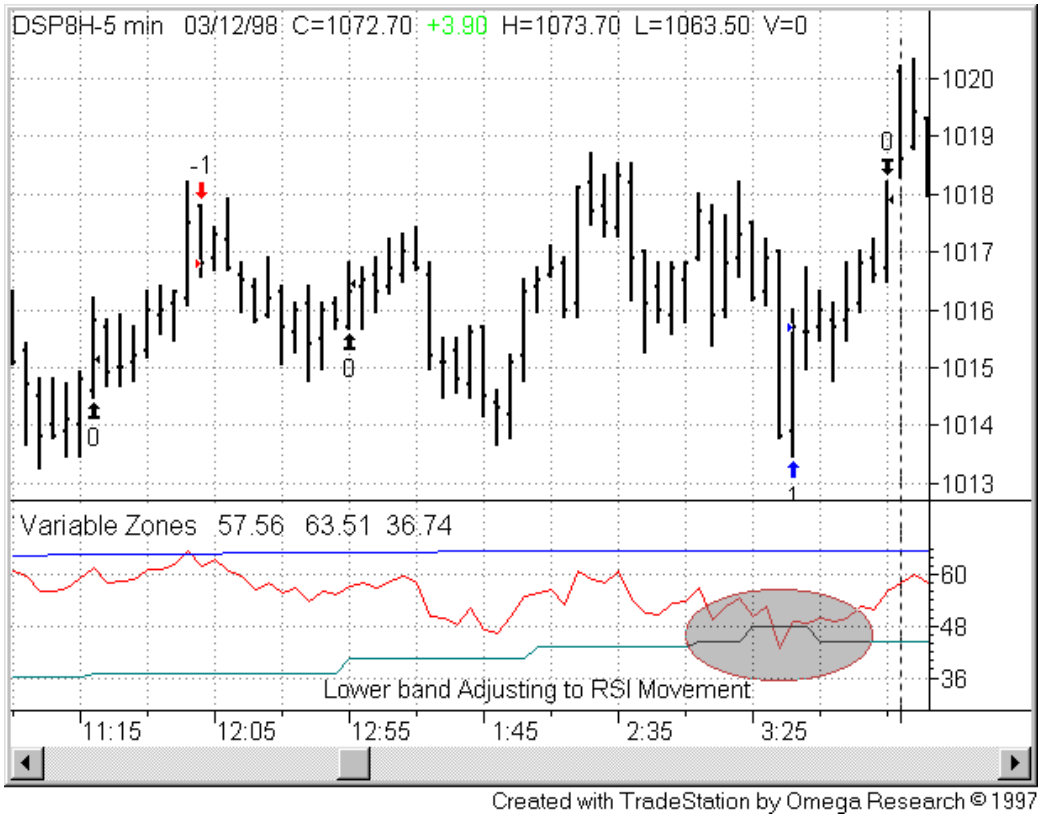


Figure 12. The Variable Zones system applied to a 5-minute chart of the S&P along with the Variable Zns Indicator

To exit from our positions, we will use a trailing stop. 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.

Also, we will use a money management stop. The charts in Figures 13 and 14 illustrate the dramatic difference between using fixed levels and variable levels. Figure 13 shows a daily chart of Intel with the Variable Zones system using fixed levels, and Figure 14 shows the same chart with the Variable Zones system using variable levels. Notice the difference in where the orders are generated.

Figure 13. Variable Zones system applied with fixed values

Figure 14. Variable Zones system applied with the variable values

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 oversold and overbought zones. The setup, entry and exits are described next.

Setup

a) Calculate the top band by using the last 4 swing highs of the RSI oscillator and determining their average. Use a strength of 5 for the swing highs, and a 20-bar RSI of the close. Then, add to the average swing high 75% of the difference between the highest swing high and the average. This is the top band.

b) Calculate the bottom band by using the last 4 swing lows of the RSI oscillator and determining their average. Use a strength of 5 for the swing lows, and a 20-bar RSI of the close. Then, subtract from the average swing low 75% of the difference between the lowest swing low and the average. This is the bottom band.

Long Entries

a) When the RSI crosses over the bottom band, then place a buy order at the close. Again, use 20 bars and the close to calculate the RSI.

Short Entries

a) When the RSI crosses under the top band, then place a sell order at the close. Again, use 20 bars and the close to calculate the RSI.

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.

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: Variable Zones (STAD3: Var Zones)

```
Input: Osc( RSI(Close, 20)), Swings(4), Strength(5), Factor(.75);
Vars: TopBand(0), BotBand(0), MP(0), StopPrice(0);

{ Setup calculations - calculating dynamic zones }
TopBand = TopVariableBand( Osc, Swings, Strength, Factor );
BotBand = BotVariableBand( Osc, Swings, Strength, Factor );

{ Entry orders }
If Osc Crosses Over BotBand then
        Buy this bar on Close;
If Osc Crosses Under TopBand then
        Sell this bar on Close;
```

```
{ Trailing stops }
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 next bar at StopPrice Stop;
        StopPrice = StopPrice + (Low - StopPrice)/3;
End;

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

Inputs

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

Inputs	Default	Description
Osc	RSI(Close,20)	The oscillator to be used and for which the overbought and oversold zones need to be calculated.
Swings	4	Number of swing highs and swing lows to use to determine the average.
Strength	5	Strength, expressed in bars, to be used to calculate swing high and swing low.
Factor	.75	Factor by which to multiply the difference to add to or subtract from the average.

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

```
Vars: TopBand(0), BotBand(0), MP(0), StopPrice(0);
```

It is always a good idea to write your calculations as functions, like we did here with the top and bottom variable bands, and like we did with the on balance true range calculation.

This way, you can easily use the calculation in another system or analysis technique.

To calculate the top and bottom bands, we use two functions we wrote called TopVariableBand and BotVariableBand. These functions are included on your CD and require four parameters, the price or value that oscillates, the number of swing highs or lows to use in determining the average, the strength of the swing highs or lows (i.e., how many bars on either side are required for the bar to be considered a swing high or swing low), and the factor by which to multiply the difference between the average and either the highest swing high or the lowest swing low.

When calculating the top band, we subtract the average from the highest swing high, multiply it by the factor and then add it to the average swing high to determine the upper band. Likewise, when calculating the bottom band, we subtract the average from the lowest swing low, multiply it by the factor and then subtract it from the average swing low to determine the lower band.

The key is that the top and bottom bands recalculate for every bar.

```
TopBand = TopVariableBand( Osc, Swings, Strength, Factor );
BotBand = BotVariableBand( Osc, Swings, Strength, Factor );
```

Long Entries

If the RSI, calculated using the close for the last 20 bars, crosses over the bottom band, then place a buy order at the close.

```
If Osc Crosses Over BotBand then
        Buy this bar on Close;
```

Short Entries

If the RSI, calculated using the close for the last 20 bars, crosses under the top band, then place a sell order at the close.

```
If Osc Crosses Under TopBand then
        Sell this bar on Close;
```

Exit Orders

First we obtain our market position using the MarketPosition function and store it in the variable MP. Then, 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.

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

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

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

Then, once we are in a long position, we will place an order to exit a long position 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 plus a third of the difference between the low and the current stop price.

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

Likewise, once we are in a short position, we will 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 minus a third of the difference between the current stop price and the high of the current bar.

```
If MP = -1 then Begin
        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 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

We can see the significant advantage of the variable bands method by using the same oscillator we use in the Variable Zones system, the RSI, but using fixed overbought and oversold levels.

The system buys the RSI crosses over the oversold level and sells short when the RSI crosses under the overbought level. When using fixed values (at 35 and 75), the system not only gives losses, but it has a huge drawdown of nearly $20,000. Figure 15 shows the System Report for the system using the fixed levels.

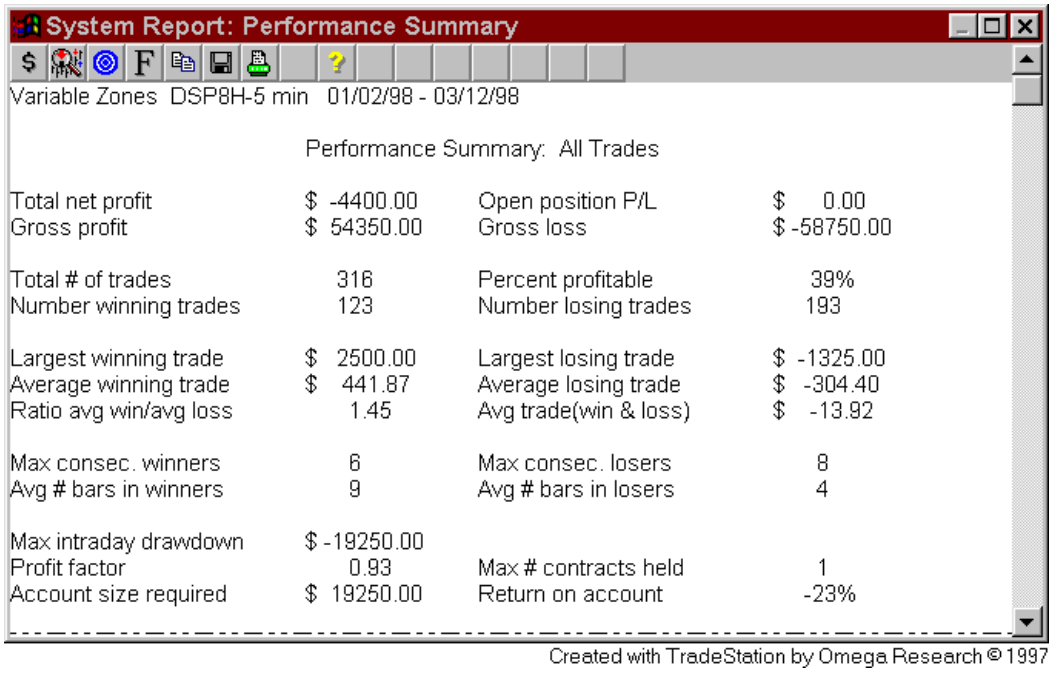


Figure 15. Sample System Report for the Variable Zones system that uses fixed values, on a daily chart of Intel

When we used the variable bands instead of the fixed levels, the system changed completely. It now shows a profit and the drawdown has been reduced to a third of the original. Figure 16 shows the System Report for the Variable Zones system.

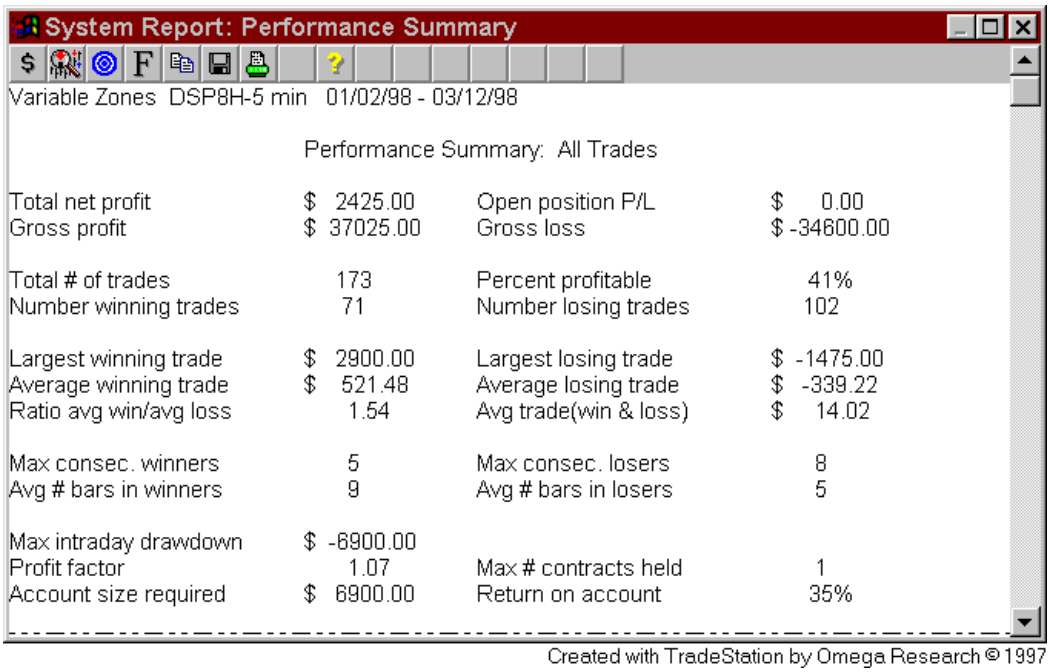


Figure 16. Sample System Report for the Variable Zones system, this time using the variable zones

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

Inside Day Breakout Failure

An Inside Day is a price bar with a high below the previous bar's high and a low above the previous bar's low. Inside Days are typically interpreted as the market being undecided towards the future direction, and they usually appear right before or immediately after a significant move.

Prices frequently continue to move higher after a breakout above the high of an Inside Day and continue to move lower after a breakout below the low of an Inside Day. This tendency is even more reliable when the breakout follows a failed breakout in the other direction on the same day. In other words, our thinking is that we have a buying opportunity when prices fall below the low of an Inside Day and then reverse on the same day to rally above the high of the Inside Day, and we have a selling opportunity when prices rally above the high of an Inside Day and then reverse on the same day to fall below the low of the Inside Day.

We decided to build a system based on this concept. The easiest way to do this is to use a chart with two data streams. First, we'll plot a 30-minute chart of the S&P, then we'll plot a daily chart of the same symbol. We first identify an Inside Day on the daily chart. Then we evaluate the 30-minute bars to monitor whether the low or the high of the Inside Day is penetrated first.

If the low of the Inside Day is penetrated first, we'll place a buy stop one tick above the high of the Inside Day. Once we enter the long position, we'll set a trailing stop one tick below the low of the Inside Day minus half of the 3-day average range of the daily bars. We'll also exit from the position at the end of the day of entry.

If the high of the Inside Day is penetrated first, we'll place a sell stop one tick below the low of the Inside Day. If filled, we'll set a trailing stop one tick above the high of the Inside Day plus half of the 3-day average range of the daily bars. We'll also exit from the position at the end of the day of entry.

Figure 1 illustrates the Inside Day chart pattern we are looking for and the buy signal based on the pattern.

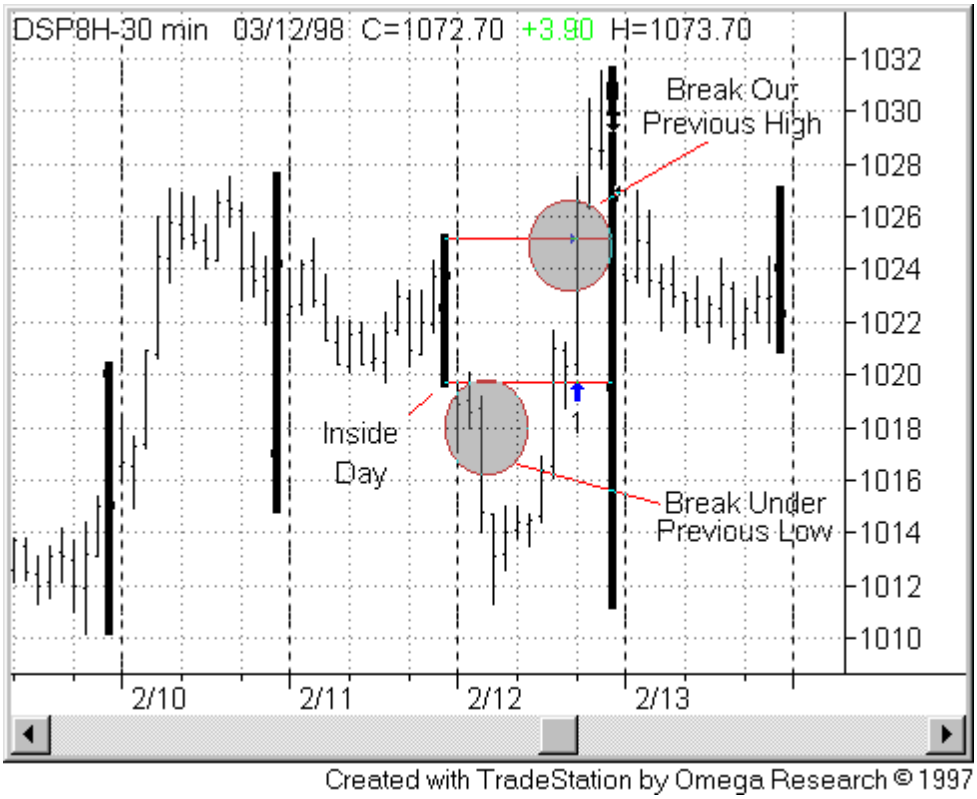


Figure 1. Inside Day chart pattern illustrated on a 30-minute S&P chart with the daily chart overlaid on top

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 identifying the Inside Day and subsequent breakouts. The setup, entry and exits are described next.

Setup

a) Identify an Inside Day, which is a price bar with a high below the previous bar's high and a low above the previous bar's low. Use the daily data to identify the Inside Day.

b) Then, on the 30-minute chart, identify where the prices fall below the low of an Inside Day or where the prices rally above the high of an Inside Day.

Long Entries

a) If the low of the Inside Day is penetrated first, we'll place a buy stop one tick above the high of the Inside Day. Therefore, if the price rallies above the high of the Inside Day, we'll buy.

Short Entries

a) If the high of the Inside Day is penetrated first, place a sell stop one tick below the low of the Inside Day. Therefore, if the price falls below the low of the Inside Day, we'll sell.

Exits

a) Once we enter the long position, we'll set a trailing stop one tick below the low of the Inside Day minus half of the 3-day average range of the daily bars. If filled, we'll set a trailing stop one tick above the high of the Inside Day plus half of the 3-day average range of the daily bars. We'll also exit from either a long or a short position at the end of the day of entry.

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: Inside Day Breakout Failure (STAD3: ID Bkout Fail)

```
Input: RngAvg(3);
Vars: InsideDay(False), DayRangeAvg(0), PlacedOrder(False), LongEntryPrice(-1), ShortEntryPrice(-1);
{ Setup calculations }
InsideDay = High of Data2 < High[1] of Data2 AND Low of Data2 > Low[1] of Data2;
DayRangeAvg = Average( TrueRange of Data2 , RngAvg ) of Data2;
If Date <> Date[1] then Begin
        PlacedOrder = False;
        LongEntryPrice = -1;
        ShortEntryPrice = -1;
End;
{ Entry setup and orders }
If InsideDay then Begin
        If PlacedOrder = False AND Low < Low of Data2 then Begin
                LongEntryPrice = High of Data2;
                PlacedOrder = True;
        End;
        If PlacedOrder = False AND High > High of Data2 then Begin
                ShortEntryPrice = Low of Data2;
                PlacedOrder = True;
        End;
```

```
        { Entry orders }
        If PlacedOrder AND LongEntryPrice <> -1 then
                Buy next bar at LongEntryPrice Stop;
        If PlacedOrder AND ShortEntryPrice <> -1 then
                Sell next bar at ShortEntryPrice Stop;
End;

{ Protective stops }
ExitLong at Low of Data2 - DayRangeAvg / 2 Stop;
ExitShort at High of Data2 + DayRangeAvg / 2 Stop;
```

Inputs

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

Inputs	Default	Description
RngAvg	3	Period, expressed in bars, used to calculate the average of the range (which is used in the calculation of protective stops)

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

```
Vars: InsideDay(False), DayRangeAvg(0), PlacedOrder(False), LongEntryPrice(-1),
ShortEntryPrice(-1);
```

We begin by identifying the Inside Day using the second symbol plotted in our chart, Data2. If the current daily bar is an Inside Day, we'll set the variable InsideDay to true. Then we calculate the 3-day average true range of the second symbol and store this in the variable DayRangeAvg. We'll use this to place our protective stops. After calculating the average true range, we check to make sure the current 30-minute bar is the first bar of the day. If it is, we set PlacedOrder to false, and LongEntryPrice and ShortEntryPrice to -1. We want to reset these variables at the beginning of each day since our system is based on one day's intra-day price movement.

```
InsideDay = High of Data2 < High[1] of Data2 AND Low of Data2 > Low[1] of Data2;
DayRangeAvg = Average( TrueRange of Data2 , RngAvg ) of Data2;
If Date <> Date[1] then Begin
        PlacedOrder = False;
        LongEntryPrice = -1;
        ShortEntryPrice = -1;
End;
```

If the current daily bar is an Inside Day, then we check to make sure we haven't yet placed an order and that the low of the current 30-minute bar is less than the low of the daily Inside Bar. If these two conditions are true, we set the LongEntryPrice to the high of the daily bar and set the variable PlacedOrder to true. We also check for the high of the current bar being higher than the high of the daily bar. If that's the case, we set the ShortEntryPrice to the low of the daily bar and set the variable PlacedOrder to true.

```
If InsideDay then Begin
        If PlacedOrder = False AND Low < Low of Data2 then Begin
                LongEntryPrice = High of Data2;
                PlacedOrder = True;
        End;
        If PlacedOrder = False AND High > High of Data2 then Begin
                ShortEntryPrice = Low of Data2;
                PlacedOrder = True;
        End;
```

Long Entries

To place a long entry, we check the variables PlacedOrder and LongEntryPrice to determine if an order should be placed (PlacedOrder = true) and we have a long entry price stored (LongEntryPrice <> -1). If so, then we will place a buy stop order on the next bar at the long entry price.

```
If PlacedOrder AND LongEntryPrice <> -1 then
        Buy next bar at LongEntryPrice Stop;
```

Short Entries

To place a short entry, we check the variables PlacedOrder and ShortEntryPrice to determine if an order should be placed (PlacedOrder = true) and we have a long entry price stored (ShortEntryPrice <> -1). If so, then we will place a sell stop order on the next bar at the short entry price.

```
If PlacedOrder AND ShortEntryPrice <> -1 then
        Sell next bar at ShortEntryPrice Stop;
```

Exit Orders

We place two trailing stop orders, one to exit any long positions at the low of the daily bar minus half the 3-day average true range, and one to exit any short positions at the high of the daily bar plus half the 3-day average true range.

```
ExitLong at Low of Data2 - DayRangeAvg / 2 Stop;
ExitShort at High of Data2 + DayRangeAvg / 2 Stop;
```

IMPORTANT: We also selected the **Close all trades at end of day session** option. This is very important and will completely change the system results if not selected. Please make sure you select this option as you apply the system to your chart.

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.

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.*

TIP: Since you are using two different data streams, the setting for the maximum number of bars the study will reference must be set for the largest data compression. In other words, if the setting is set to 3 and you are using daily and minute bars, then 3 daily bars are necessary to begin calculating the system, regardless of how many minute bars are already on the chart.

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.

Notes: *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

This pattern is one of those very-difficult-to-test ideas as the pattern occurs very infrequently, making it difficult to come up with statistically sound results. When we apply this to a 6 month

30-minute chart of the S&P futures contract, we only get 6 trades (so the bar pattern has appeared about once a month). However, 5 times out of the 6 the system has been able to capture profits from the trade. This is definitely a pattern to keep an eye out for, but you may not be able to generate trades frequently enough in order to produce constant revenues.

Suggestions for Improvement

A possible improvement can be to note the time on which the breakout of the high and the low of the bar occur. The S&P tends to trend strongly in the last minutes of every day session, so the later the second breakout occurs, the more reliable the pattern might be. Note that all markets are different from one another, so this might not apply to other markets.

Standard Deviation – Volatility

Standard deviation is a statistical measure of how 'normal' the prices are. It is a number that basically expresses how distant the prices are from the mean (or average). One of the characteristics of most volatility measurements is that they are non-directional; in other words, measuring volatility will tell you the market is active, but it will not tell you the direction in which it is moving.

When developing a volatility breakout system, we want to try to identify when the market has made a significant move, either up or down, in an effort to get in the market and ride the momentum to a quick profit. If the volatility of the market is high, the market is making big moves on a bar by bar basis in an uncertain direction, and in order to deem a move in some direction significant, it has to be a greater move than when the market has low volatility.

Figure 2 shows a 5-minute chart of the S&P with the Standard Deviation Range Indicator applied (STAD3: Std Dev Rg). We wrote this indicator so we could more easily visualize the relationship between volatility and price movement.

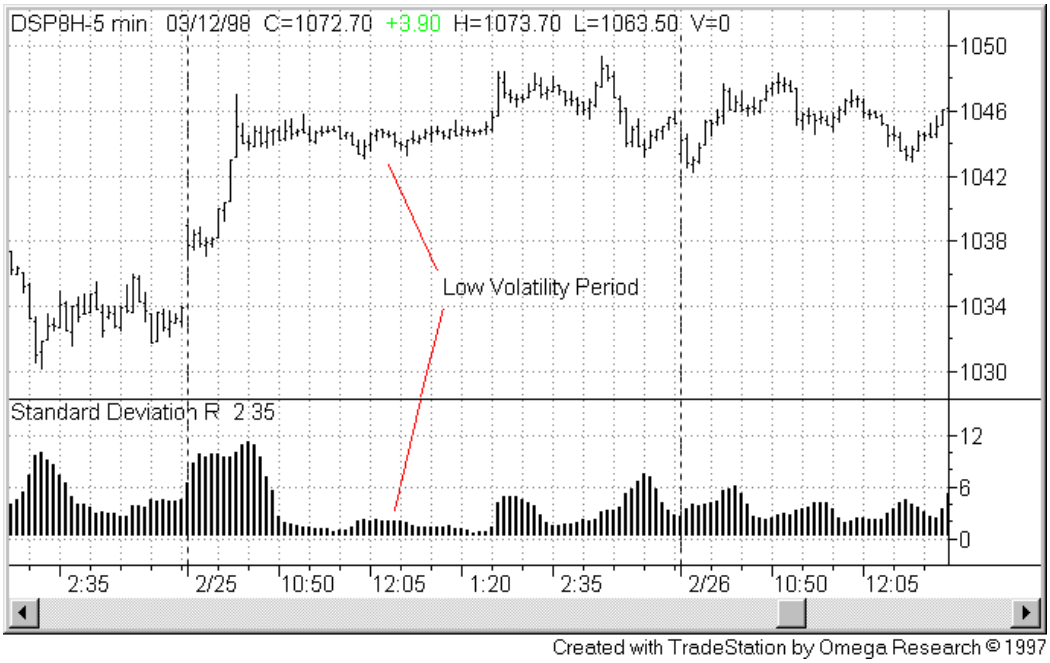


Figure 2. Five-minute chart of the S&P with the Standard Deviation Range Indicator applied

With this idea in mind, we decided to try to find long and short entry prices that adjust according to the recent statistical volatility level of the security we're trading. The higher the volatility, the further away from the current price we want to put our entry points.

In order to accomplish this, we will define an initial upper and lower band by calculating 2 standard deviations above and below the 9-bar average price of the close. We will then divide this band range by the 4-bar average of the close and add to 1 for the upper band and subtract from 1 for the lower band. The general idea is as follows:

1 + (Band range / 4-bar Average of the Close)
1 – (Band range / 4-bar Average of the Close)

This gives us, expressed as a percentage (for example, 1.02 and .98 with the bandwidth at 2% of the prices), the relationship between volatility and the price. Finally, we multiply each of these values by the close of the current bar to obtain our long and short entry points.

1 + (Band range / 4-bar Average of the Close) * Close
1 – (Band range / 4-bar Average of the Close) * Close

As you look at the formula, you will notice that the higher the distance between the standard deviation lines (in other words the higher the volatility, the further away the bands will be).

Once we establish our positions, we will exit at the first profitable open as well as place a money management stop, the amount of which will be based on what we're trading.

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 standard deviation of the price and the entry points. The setup, entry and exits are described next.

Setup

a) Calculate 2 standard deviations above and below the average price; these are the initial upper and lower bands. Divide the range of the bands by the 4-bar average of the close, then add this number to 1 to obtain the upper band and subtract from 1 to obtain the lower band.

Long Entries

a) Multiply the two values obtained in the setup by the close of the current bar to obtain the long and short entry points. Place a buy stop order at the long entry price.

Short Entries

a) Multiply the two values obtained in the setup by the close of the current bar to obtain the long and short entry points. Place a sell stop order at the short entry price.

Exits

a) Exit at the first profitable open, which is calculated by comparing the open with the entry price (accounting for a desired profit expressed in points and commission).

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: Standard Deviation - Volatility (STAD3:Std Dev - V)

```
Input: Price(Close), StdDv(2), Length(9), ProfitPt(0);
Vars: TopBand(0), BotBand(0), BandRange(0), SellPrice(0), BuyPrice(0);
```

```
{Setup calculations - calculating bands & buy and sell prices }
TopBand = Average(Close, Length) + StdDev(Price, Length) * StdDv;
BotBand = Average(Close, Length) - StdDev(Price, Length) * StdDv;
BandRange = TopBand - BotBand;
BuyPrice = Close  * ( 1 + BandRange / Average(Close, 4) );
Sellprice = Close  * ( 1 - BandRange / Average(Close, 4) ) ;

{ Entry orders }
Buy  next bar at BuyPrice Stop;
Sell  next bar at SellPrice Stop;

{ Exit at first profitable open }
If MarketPosition = 1 and Open next bar > EntryPrice + ProfitPt + Commission *
        CurrentContracts / BigPointValue then ExitLong ("Long Profit") next bar at Open;
If MarketPosition = -1 and Open next bar < EntryPrice - ProfitPt - Commission *
        CurrentContracts / BigPointValue then ExitShort ("Short Profit") next bar at Open;
```

Inputs

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

Inputs	Default	Description
Price	Close	Price to use in calculating the standard deviation and on which to base the entry points
StdDv	2	Number of standard deviations used to calculate the upper and lower bands
Length	9	Period, expressed in bars, used to calculate the standard deviation as well as the average of the close used to establish the initial bands
ProfitPt	0	Number of points desired in profit before exiting position

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

```
Vars: TopBand(0), BotBand(0), BandRange(0), SellPrice(0), BuyPrice(0);
```

We begin by calculating the top and bottom bands. For the top band, we calculate the 9-bar average of the close and add it to twice the standard deviation of the last 10 closing prices. For the bottom band, we calculate the same 9-bar average but subtract from it twice the standard deviation. Once we have the bands, we calculate their range by subtracting the bottom band from the top band.

```
TopBand = Average( Close, Length ) + StdDev( Price, Length ) * StdDv;
BotBand = Average( Close, Length ) - StdDev( Price, Length ) * StdDv;
BandRange = TopBand - BotBand;
```

Then, we obtain the buy and sell prices by dividing the band range by a 4-bar average of the close and then adding or subtracting this from 1, respectively. Then we multiply the resulting number by the close.

```
BuyPrice = Close  * ( 1 + BandRange / Average(Close, 4) );
Sellprice = Close  * ( 1 - BandRange / Average(Close, 4) ) ;
```

Long Entries

Once we calculate the buy price, we place a buy stop order at this price.

```
Buy next bar at BuyPrice Stop;
```

Short Entries

Once we calculate the sell price, we place a sell stop order at this price.

```
Sell next bar at SellPrice Stop;
```

Exit Orders

This system will exit from any position at the first profitable open. To determine when the first profitable open occurs, we compare the open of the next bar to the entry price, making sure we take into account commissions.

Commissions are specified on a per contract/share basis; therefore, we have to multiply the value returned by the Commission function by the number of contracts held (returned by the CurrentContracts function), and then divide this value by the big point value of the instrument. The big point value is the dollar value of a full point move; we can easily obtain it using the BigPointValue function:

```
If MarketPosition = 1 and Open next bar > EntryPrice + ProfitPt + Commission *
        CurrentContracts / BigPointValue then ExitLong ("Long Profit") next bar at Open;
If MarketPosition = -1 and Open next bar < EntryPrice - ProfitPt - Commission *
        CurrentContracts / BigPointValue then ExitShort ("Short Profit") next bar at Open;
```

We'll also use a money management stop, as discussed in the next section.

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 100 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

This system (when using a ProfitPt input set at 2) shows the system results of a typical volatility expansion system. Figure 3 shows the System Report. It has a high percentage of winning trades (slightly over 72%), and an average winning versus losing trades ratio (near 1.00).

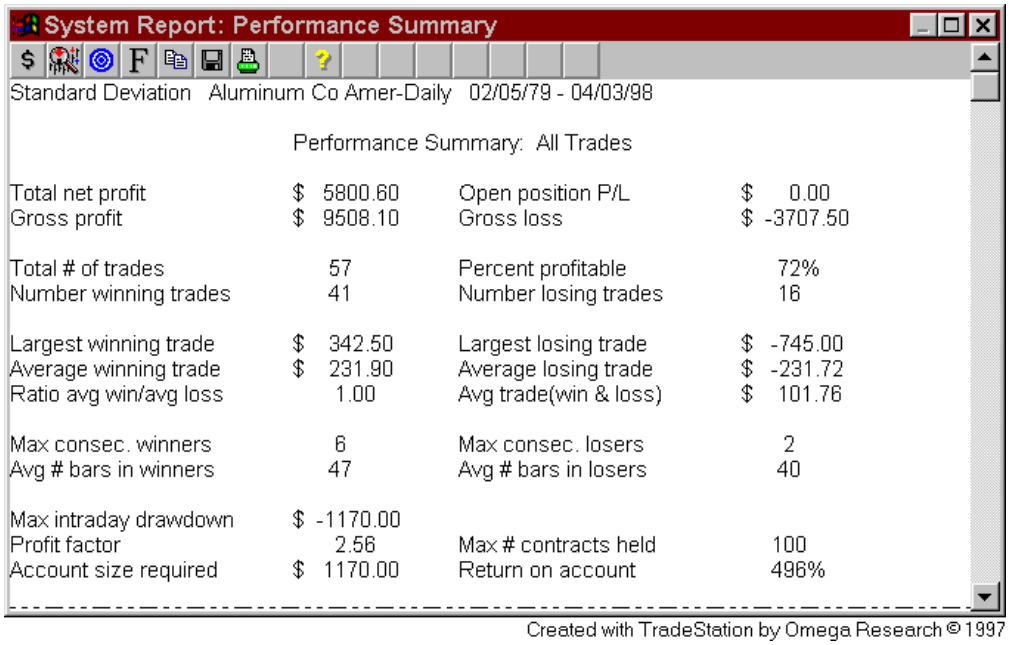


Figure 3. Sample System Report for the Standard Deviation - Volatility system

Suggestions for Improvement

This system really has very little room for improvement. However, after looking at the drawdown, which is somewhat high considering that we have a maximum of 2 consecutive losing trades and an average losing trade of around $230, we decided to look at the Maximum Adverse Excursion graph, available in Portfolio Maximizer (Figure 4).

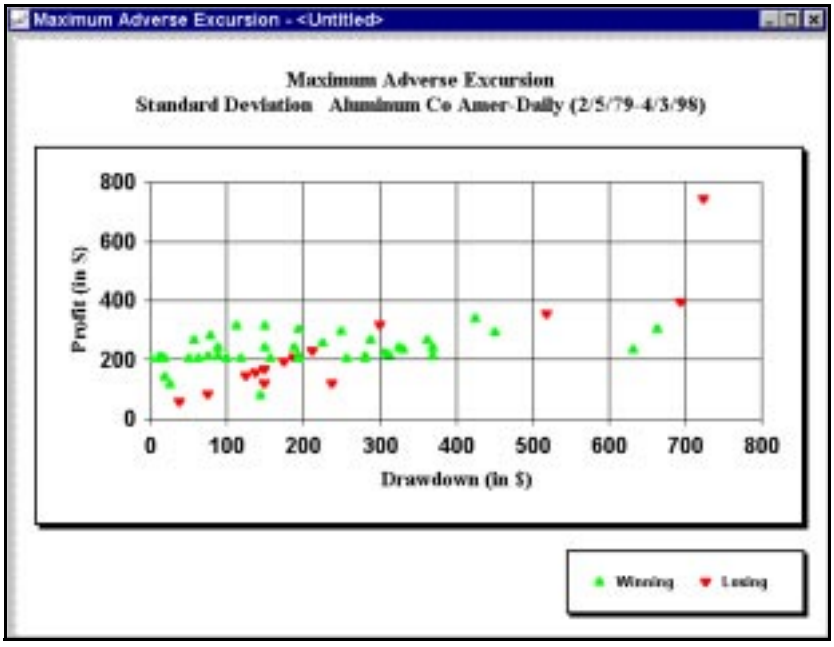


Figure 4. Maximum Adverse Excursion graph for the Standard Deviation - Volatility system

The analysis in this case is not as clear cut as with other systems. While we have three losing trades with over $500 maximum drawdown, there are also two winning trades of around $300 each. Placing a stop at $500 would both reduce our drawdown AND affect our profits, because we would be converting these two winning trades into losers. In this type of situation, we need to choose between slightly hurting our profits and reducing our drawdown.

Volatility Clusters

One assumption that is frequently made when studying volatility breakouts is that on average there will be one bar that has high volatility per every *x* number of bars with normal volatility. This assumption is illustrated in a bar chart in Figure 5.

Figure 5. Commonly assumed volatility pattern

In the above example, on average we will have one period of high volatility for every six normal periods.

However, in reality, high volatility periods tend to cluster as groups, and one period of high volatility will usually be followed by another of more or less the same magnitude. The reason is that whatever event that caused that one period of high volatility will not cease to have influence over the market simply because we have started drawing a new bar on our charts. In mathematical terms, we are taking a indiscrete event (like the influence of a news story over a stock price) and representing it in discrete terms (bars). We propose that high volatility periods will tend to look like the chart in Figure 6.

Figure 6. Realistic volatility pattern

And as explained in the previous system, Standard Deviation - Volatility, when we study volatility, we are not considering in any way the direction of the market. Volatility is a function of how much the market is moving, not of where it is going.

For the purposes of this system, we are going to look for high volatility in two ways, either as periods where the bar range increases significantly from its average or as periods when the distance from the current close to the previous close increases significantly from the average distance between two consecutive closes.

Because we are looking for increase in volatility, we will only look for the move to be greater than 2 standard deviations above the average move. If this happens, we will place stop orders above and below the market to enter the market long and short, respectively.

To determine the long and short entry prices, we will calculate the 3-bar average of the range and add and subtract it from the close, respectively. Because the range of the current bar is the one we specified as being greater than the average, we will use the 3-bar average of the range

of 1 bar ago; this way we will leave the bar with the volatility spike out of the entry price calculation.

We will exit at the first open that meets our profit target, and we will place a money management stop to protect ourselves against losing trades.

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 standard deviations, averages and ranges in order to find the entry prices. The setup, entry and exits are described next.

Setup

a) Calculate the difference between the close of the current bar and the close of the previous bar. Find the 18-bar average of the difference between two consecutive closes. Add to this 2 standard deviations of the difference of the closes over 18 bars.

b) Calculate the 18-bar average of the range and add to this 2 standard deviations of the range over the last 18 bars.

Long Entries

a) Compare the difference between the two consecutive closes to the standard deviation calculation from the setup. It the difference is greater, place a buy stop at the close plus the 3-bar average of the range (1 bar ago).

b) Compare the range to the standard deviation calculation from the setup. If the range is greater, place a buy stop at the close plus the 3-bar average of the range (1 bar ago).

Short Entries

a) Compare the difference between the two consecutive closes to the standard deviation calculation from the setup. It the difference is greater, place a sell stop at the close minus the 3-bar average of the range (one bar ago).

b) Compare the range to the standard deviation calculation from the setup. If the range is greater, place a sell stop at the close minus the 3-bar average of the range (one bar ago).

Exits

a) Exit at the first profitable open, which is calculated by comparing the open with the entry price (accounting for a desired profit expressed in points and commission).

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: Standard Deviation - Volatility (STAD3: Volty Clustrs)

```
Inputs: Stdev(2), Length(18), ProfitPt(0);
Vars: CloseToClose(0), CTCStdDev(0), RangeStdv(0), BigVolatility(False);

{ Setup calculations }
CloseToClose = AbsValue(Close - Close[1]);
CTCStdDev = Average(CloseToClose, Length) + StdDev(CloseToClose, Length) * Stdev;
RangeStdv = Average(Range, Length) + StdDev( Range, Length ) * Stdev;
BigVolatility = CloseToClose > CTCStdDev[1] OR Range > RangeStdv[1];
```

```
{ Entry orders }
If BigVolatility then Begin
        Buy next bar at Close + Average(Range,3)[1] Stop;
        Sell next bar at Close - Average(Range,3)[1] Stop;
End;

{ Exit at first profitable open }
If MarketPosition = 1 and Open next bar > EntryPrice + ProfitPt + Commission *
        CurrentContracts / BigPointValue then ExitLong ("Long Profit") next bar at Open;
If MarketPosition = -1 and Open next bar < EntryPrice - ProfitPt - Commission *
        CurrentContracts / BigPointValue then ExitShort ("Short Profit") next bar at Open;
```

Inputs

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

Inputs	Default	Description
Stdev	2	Number of standard deviations used to compare with the values to determine high volatility
Length	18	Period, expressed in bars, used to calculate the averages and standard deviation
ProfitPt	0	Number of points desired in profit before exiting position

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

```
Vars: CloseToClose(0), CTCStdDev(0), RangeStdv(0), BigVolatility(False);
```

We begin by calculating the difference of the close of the current bar and the close of the previous bar. We use the AbsValue function so the difference is always a positive number. We store this value in the variable CloseToClose. Then, we calculate the average of the difference over 18 bars and add to this value 2 standard deviations of the difference over 18 bars. We store the resulting value in the variable CTCStdDev. This becomes one of our benchmarks to find high volatility.

We also calculate the average of the range over 18 bars and add to this value 2 standard deviations of the range over 18 bars. We store the resulting value in the variable RangeStd, and this becomes our second benchmark to find high volatility.

Then, we compare the difference between the last two consecutive closes to the average plus 2 standard deviations. We also compare the current range to the average plus the standard deviation of the range. If either of the current values is greater than the average plus the standard deviation calculation, we consider this to be a sign of high volatility so we set BigVolatility to true.

```
CloseToClose = AbsValue(Close - Close[1]);
CTCStdDev = Average(CloseToClose, Length) + StdDev(CloseToClose, Length) * Stdev;
RangeStdv = Average(Range, Length) + StdDev( Range, Length ) * Stdev;
BigVolatility = CloseToClose > CTCStdDev[1] OR Range > RangeStdv[1];
```

Long Entries

If BigVolatility is true (therefore, there is high volatility), we place a buy stop order at the close plus the 3-bar average of the range (1 bar ago).

```
If BigVolatility then Begin
        Buy next bar at Close + Average(Range,3)[1] Stop;
        Sell next bar at Close - Average(Range,3)[1] Stop;
End;
```

Short Entries

If BigVolatility is true (therefore, there is high volatility), we place a sell stop order at the close minus the 3-bar average of the range (1 bar ago).

```
If BigVolatility then Begin
        Buy next bar at Close + Average(Range,3)[1] Stop;
        Sell next bar at Close - Average(Range,3)[1] Stop;
End;
```

Exit Orders

This system will exit from any position at the first profitable open. To determine when the first profitable open occurs, we compare the open of the next bar to the entry price, making sure we take into account commissions.

Commissions are specified on a per contract/share basis; therefore, we have to multiply the value returned by the Commission function by the number of contracts held (returned by the CurrentContracts function), and then divide this value by the big point value of the instrument. The big point value is the dollar value of a full point move; we can easily obtain it using the BigPointValue function:

```
If MarketPosition = 1 and Open next bar > EntryPrice + ProfitPt + Commission *
        CurrentContracts / BigPointValue then ExitLong ("Long Profit") next bar at Open;
If MarketPosition = -1 and Open next bar < EntryPrice - ProfitPt - Commission *
        CurrentContracts / BigPointValue then ExitShort ("Short Profit") next bar at Open;
```

We'll also use a money management stop, as discussed in the next section.

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.

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 System Report for this system (with a ProfitPt of 1.5 points) is typical of a volatility breakout system, as shown in Figure 7. It has a high percentage of winning trades at 60% and an average winning versus losing trades ratio slightly under 1(.84).

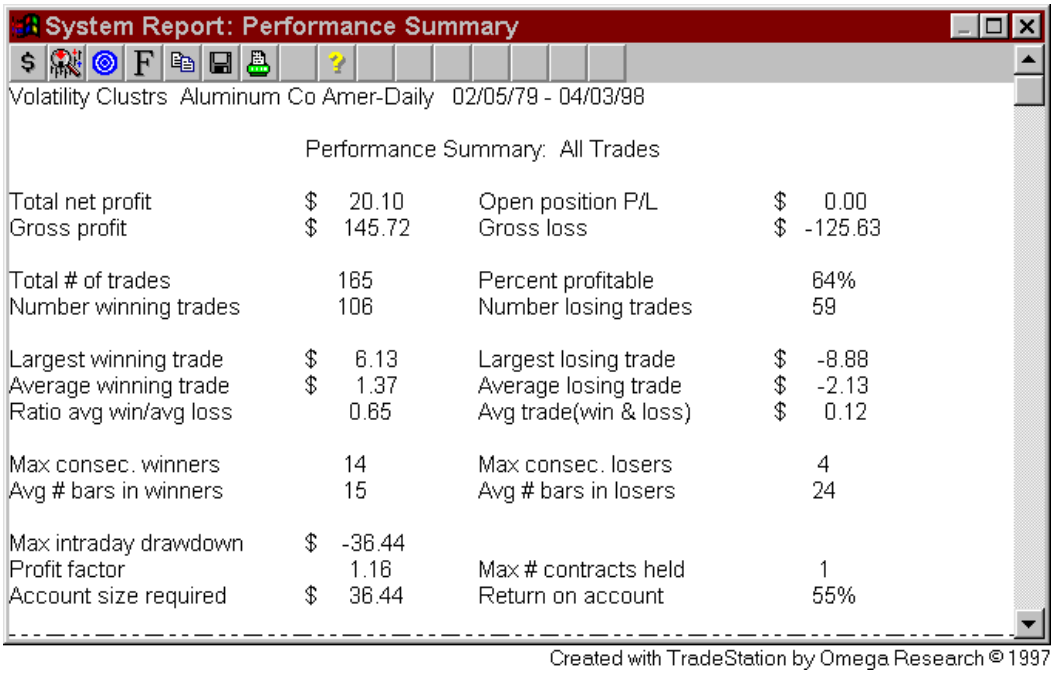


Figure 7. Sample System Report for the Volatility Clusters system

Suggestions for Improvement

This system may trade too often and therefore be somewhat difficult to trade. In order to 'slow it down' we may want to use a filter that would further indicate a significant move, for example, an increase of volume over the average volume, or divergence between the on balance true range and the price. Such a change also will probably improve the percentage of winning trades.

APPENDIX A

Volume in Review

Each volume of the *System Trading and Development Club* provides feedback on the previous volume. This appendix provides feedback on Volume 2.

Feedback can include general comments from clients as well as answers to specific questions clients may have asked regarding one of the systems. This appendix will also address any discrepancies or ommissions, if any, made from the systems in 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

STAD Club Volume 2 contained a section in Chapter 4 called, "Determining Short-Term Market Outlook" (page 75). The section presented 3 systems and discussed the use of systems in helping you determine the short-term market outlook by providing you with statistical information. Two of the three systems discussed in the section were contributed by Joe Krutsinger. We apologize for not giving due credit in the previous manual.

Joe is a Commodity Trading Advisor, registered with the CFTC and President both of Joe Krutsinger, Inc., a consulting firm, and Krutsinger & Krutsinger, Inc. an Omega Solution Provider and publisher of PORTANA-The Portfolio Analyzer.

He has constructed dozens of futures trading systems over the years for his own clients and other traders. Joe has authored and co-authored multiple books, including *The Trading Systems ToolKit*, and is a popular speaker at futures industry trading conferences.

General Feedback

Carlos Lourenco wrote:

Hi:

Enjoying the 2 system volumes so far.

I'm having lots of problems with multi-data/2 time-frame type of systems and was wondering if you'd consider developing/showcasing something for users to study. I'm having trouble with a system that used 30-minute bars on Data1 and daily bars on Datas 2-6.

Hope you'll take me up on this request.

Thanks,

Carlos Lourenco

Omega Research: *Please take a look at Chapter 4, Volatility Breakout Systems. The Inside Day Breakout Failure system is an intra-day system based on 2 data streams. Data1 is a 30-minute chart, and Data2 is the same symbol plotted daily.*

Opening Gap System

The following e-mail was received from Peter Perry:

Folks,

First, the club is first rate! Keep up the quality work.

Second, while trying to use the volatility breakout sysytem "Opening Gap," the one using Data1 and Data2, I had difficulty getting the system to work. Finally, after a few calls to the Omega Research Technical Support Department, we realized that the Max Bars Back setting in the Properties [tab] must be set to the SMALLEST number of bars of the two "Datas" (e.g., Data2 in this case) or else the system (and any indicators, ShowMe studies, etc.) vanish! I could not find a discussion of this phenomenon anywhere in the documentation. Perhaps others have had this problem too.

Third, would you consider hosting an Internet site where club members could discuss and exchange ideas and experiences relating to the systems you provide—someting similar to the www.realtraders.com site.

Cheers,

Peter Perry

Omega Research: *We are in the process of creating such a site, actually a newsgroup, where club members can do just that—exchange ideas and experiences relating to the systems we provide. While we will announce it as soon as it's ready, keep an eye on our web site for up-to-the-minute details... Thanks for the feedback!*

Regarding the Opening Gap System, we apologize for not making it clear that when you are dealing with multiple data streams, the maximum number of bars to reference refers to all data streams. For example, say you have a chart with 3 symbols, one is daily, one is 30-minute and the other is 5-minute. Also assume you are plotting an indicator based on the 3 symbols and its maximum number of bars to reference setting is set to 10. In this case, the analysis technique will not plot until there are 10 daily bars.

INDEX

S

T

V

W