OMEGA RESEARCH

SYSTEM TRADING and DEVELOPMENT CLUB

VOLUME 6

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

INTRODUCTION

Welcome to Volume 6

Welcome to Volume 6 of the Omega Research System Trading and Development Club, the revolutionary learning tool offered by Omega Research to help you make the most of your trading potential.

This sixth volume of the Omega Research System Trading and Development Club contains 9 new systems and a "Learning More" chapter that we've created to help you get started developing your own systems.

The systems in Volume 6 incorporate popular concepts such as Accumulation Swing Index, ADX, Commodity Channel Index, Parabolic, Range Leaders and Volume-Weighted Momentum. 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 9 systems and the "Learning More" chapter, 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 our trading rules, to writing them in EasyLanguageTM, 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:

- Chapter 1: General System Development Concepts
- Chapter 2: Trending Systems
- Chapter 3: Support & Resistance Systems
- Chapter 4: Volatility Breakout Systems
- Chapter 5: Learning More About System Trading & Development
- Appendix A: Volume in Review
- Index

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 305-485-7000) and ask about the following educational services:

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 PowerEditorTM.

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 **STAD6.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 STAD6. 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) 485-7598**

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.

Please realize that 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. Therefore, if you need technical support on EasyLanguage, please use the above fax number or 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.

CHAPTER 1:

General System Development Concepts

In this General System Development Concepts chapter, we will focus on the benefits of systems trading, how to get ideas for trading systems, and the ten-step process of building winning systems.

In This Chapter

Benefits

There are at least five major benefits of trading in a systematic manner as opposed to trading in a discretionary manner:

1. You'll have a system that is compatible with your own personality and trading style - a system that you are comfortable with and that you can follow.

2. You will eliminate overly emotional trading and reduce the stress of constantly making subjective, spur-of-the-moment trading decisions.

3. You will have objective entry and exit criteria that have been validated by historical testing of quantifiable data.

4. You will know the maximum peak-to-valley drawdown that your system has experienced in the past, and you can make sure that you are adequately capitalized (both financially and psychologically) to withstand another worst-case drawdown.

5. You will gain confidence in both your system and yourself, thus strengthening your ability to follow your system and to trade in a highly disciplined manner.

As you continue to become more proficient as a systems trader, you will almost certainly discover even more benefits of a systematic approach.

Getting Ideas For Systems

We can easily think of at least five great ways to get ideas for trading systems. You'll probably come up with at least a few more. Here's our quick list:

1. SuperCharts' and TradeStation's built-in indicators, ShowMe™ studies, PaintBars™, and systems

2. Trading As A Business by Charlie Wright (available from Omega Research)

3. Jack Schwager's Complete Guide to Designing and Testing Trading Systems (12 videos, CD, manual; available from Omega Research)

4. OmegaWorld (May, 1999, Caesars Palace, Las Vegas)

5. And, of course, Omega System Trading & Development Club (ten new trading ideas with manual and CD, published six times per year. Club members also receive a password for Omega's STAD Club online forum.)

Once we're convinced that systems trading is more likely to generate consistent profits than discretionary trading, and once we have an idea for a trading system, how do we progress from an idea to a complete system?

We hope the following ten-step plan will prove useful:

1. Write your trading idea as a ShowMe study. Scroll through several years of data to develop a sense of how your idea performs.

2. Write a very simple system based on your idea. For example, you could write a system that enters a position based on your idea and exits the position automatically after n-days. Alternatively, you could write a stop-and-reverse system that uses your idea to enter, exit, and reverse positions.

3. Design a setup for your system. A setup alerts you that a trading opportunity has developed. Setups don't get you into a trade, but they do tell you that market conditions have become favorable for a trade. An example of a buy setup is a market posting two consecutive closes above a moving average. An example of a sell setup is the Relative Strength Index (RSI) crossing from above 70 to below 70.

4. Design an entry for your system. An entry is the criterion that must be met after a setup for a trade to be initiated. An example of a buy entry is a market rallying one average daily range above yesterday's close. An example of a sell entry is a market's decline below the previous week's low.

5. Design an exit for your system. An exit is the criterion by which a trade is closed out. Trailing stops, profit targets, and exit conditions will account for most of your system's exits.

A trailing stop is set below the current price for a long position and above the current price for a short position. When you are in a long position, you raise the trailing stop as the market trades higher to lock in profits; while short, you lower the trailing stop as the market trades lower, locking in profits.

An alternative to exiting on a trailing stop is exiting at a profit target. A profit target closes out a trade when the price reaches a specified objective. One example of a profit-target exit is to close out a position on the second close above the high of the entry day. Another example is to automatically close out a trade when open profits equal three times the initial risk on the trade.

An exit condition gets you out of a trade when a market no longer justifies an open position. Good traders do not always rely on stops to exit their trades. If the technical condition that got you into a trade (e.g. a rising moving average) is no longer in effect, you should exit the trade immediately rather than waiting for your stop to be hit.

6. Select the data on which you will test your system. For example, you might choose to test your system on continuous, back-adjusted data on U.S. Treasury Bonds from January, 1978 through December, 1997.

7. Divide the test data into five equal parts. Since you are going to test your system on 20 years of data, each part consists of four years. The first four years (01/02/78 - 12/31/81) are reserved for the backward test, and the last four years (01/02/94 - 12/31/97) are reserved for the forward test. The middle 12 years (01/02/82 - 12/31/93) are the data on which you will test and optimize your system.

8. Test and optimize your system on the large, middle section of data. To evaluate the results of testing and optimizing, you should consider several factors including equity curve, net profit, percent profitable, profit factor (dollars won per dollar lost), average trade, and maximum drawdown.

9. Backward and forward test your system on the out-of-sample data you reserved. The test results will probably not be as good as the results on the data for which your system was optimized. However, for your system to be tradable, the backward and forward tests should yield favorable results. Your system is unlikely to perform better in the future than it did on the out-of-sample data. Check your system's performance on the same key factors that you evaluated during your test of the sample data (equity curve, net profit, percent profitable, profit factor, average trade, and maximum drawdown).

10. Trade your system with consistency, confidence, and courage.

Now, let's build a winning trading system from the ground up.

Chart 1 applies ten-bar high and ten-bar low ShowMe studies to a daily chart of the Deutsche Mark. Going long at points A, C, and E, and going short at points B and D seems like a promising idea.

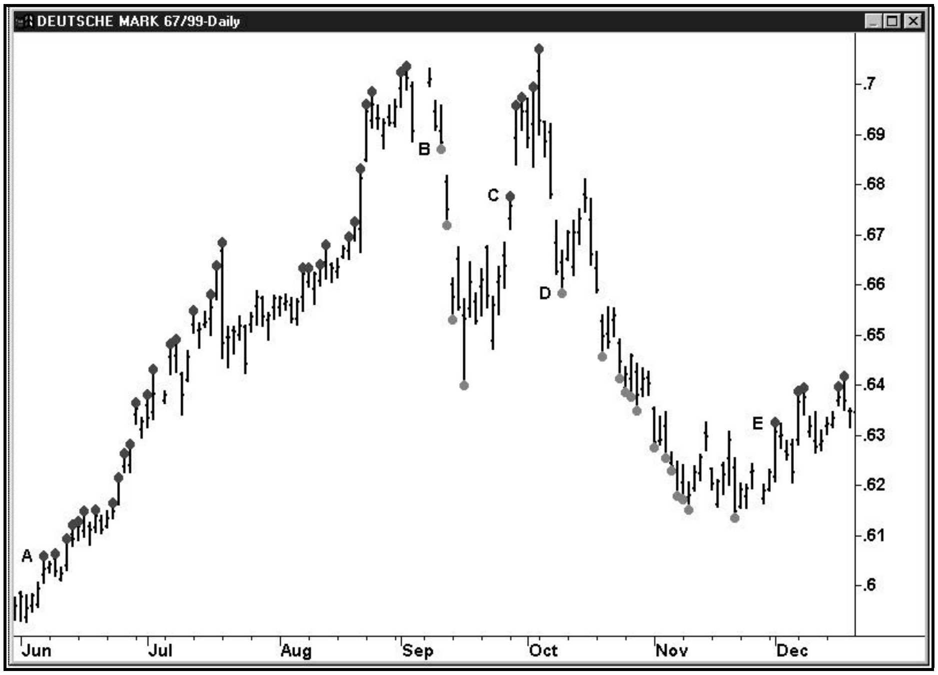
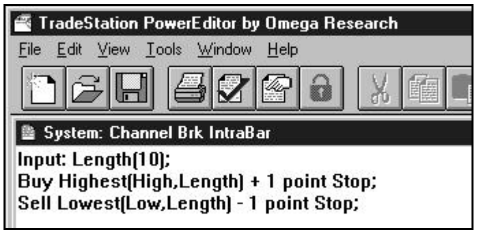


Chart 2 is Omega's EasyLanguage code for a simple stop-and-reverse ten-bar channel breakout system. Let's apply this system to the D-Mark.


TradeStation PowerEditor by Omega Research
File
Edit
View
Tools
Window
Help
System: Channel Brk IntraBar

```
Input: Length(10);
Buy Highest(High,Length) + 1 point Stop;
Sell Lowest(Low,Length) - 1 point Stop;
```

Chart 3 shows five trades generated by the system. Up arrows indicate new longs; down arrows indicate new shorts. The small flags to the left of the entry bars show the price at which the trade was entered.

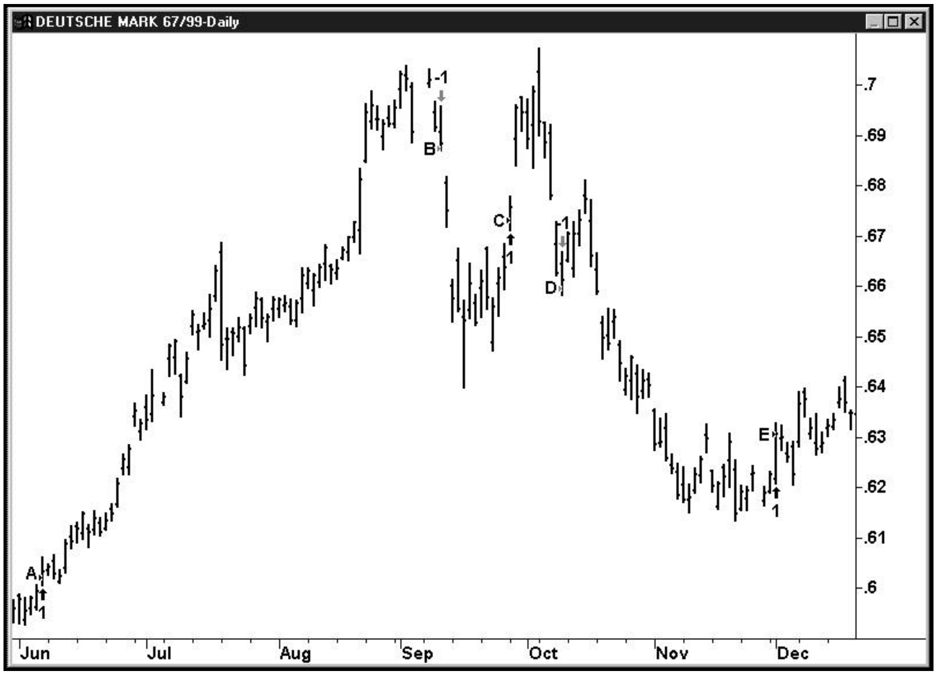


Chart 4 is the system's equity curve over approximately 150 trades. It looks encouraging, so let's see if we can make it even better.

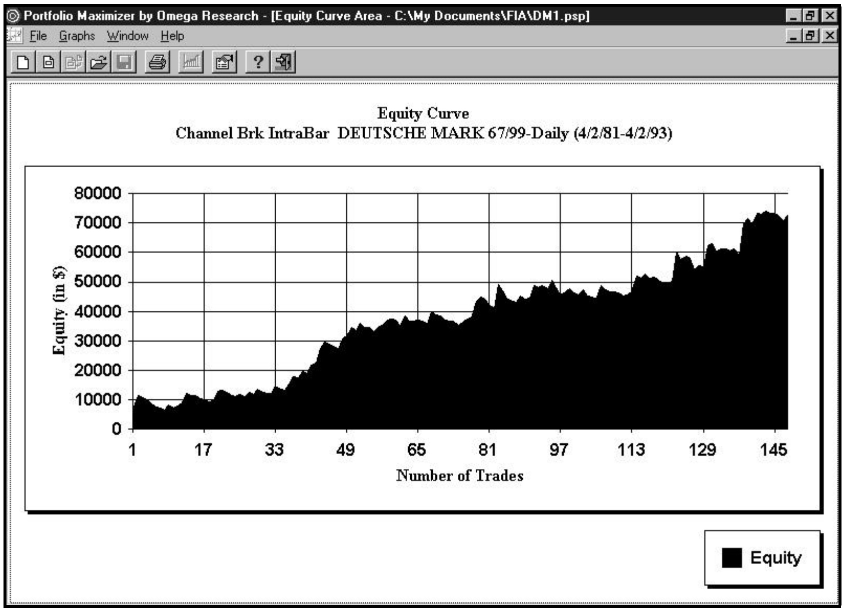
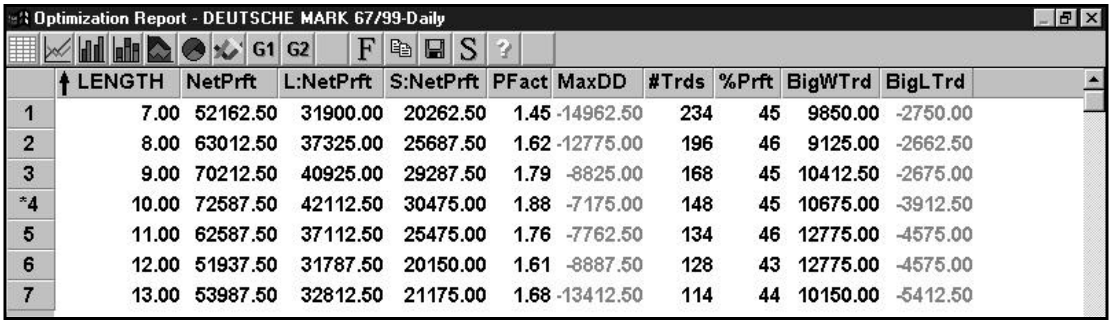

Portfolio Maximizer by Omega Research - [Equity Curve Area - C:\My Documents\FIA\DM1.psp]
File Graphs Window Help
Equity Curve
Channel Brk IntraBar DEUTSCHE MARK 67/99-Daily (4/2/81-4/2/93)
80000
70000
60000
50000
40000
30000
20000
10000
0
Equity (in $)
1
17
33
49
65
81
97
113
129
145
Number of Trades
Equity

Chart 5 is the optimization report for channel length. We tested the values seven through 13 in increments of one. The ten-bar channel (which was our software's default value) produced the best results.



	LENGTH	NetPrft	L:NetPrft	S:NetPrft	PFact	MaxDD	#Trds	%Prft	BigWTrd	BigLTrd
1	7.00	52162.50	31900.00	20262.50	1.45	-14962.50	234	45	9850.00	-2750.00
2	8.00	63012.50	37325.00	25687.50	1.62	-12775.00	196	46	9125.00	-2662.50
3	9.00	70212.50	40925.00	29287.50	1.79	-8825.00	168	45	10412.50	-2675.00
*4	10.00	72587.50	42112.50	30475.00	1.88	-7175.00	148	45	10675.00	-3912.50
5	11.00	62587.50	37112.50	25475.00	1.76	-7762.50	134	46	12775.00	-4575.00
6	12.00	51937.50	31787.50	20150.00	1.61	-8887.50	128	43	12775.00	-4575.00
7	13.00	53987.50	32812.50	21175.00	1.68	-13412.50	114	44	10150.00	-5412.50

Chart 6 shows that all the values from seven to 13 produced good results, and that ten was the best.

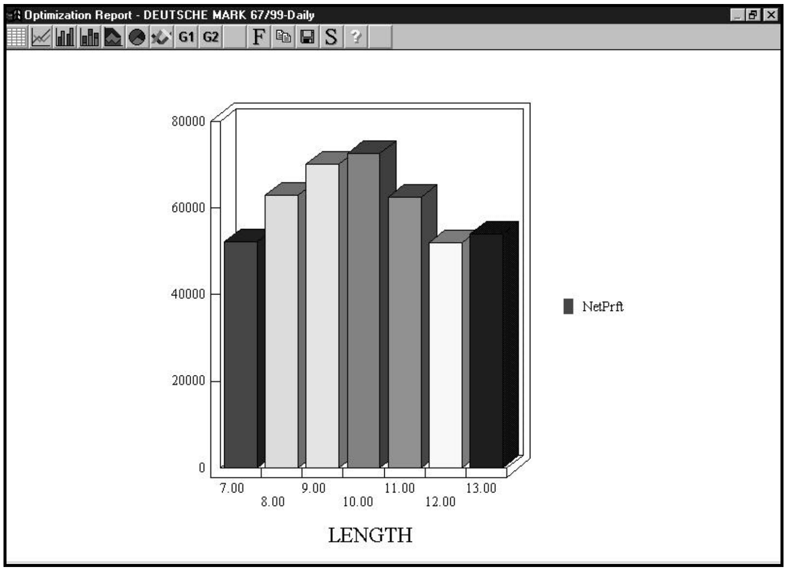
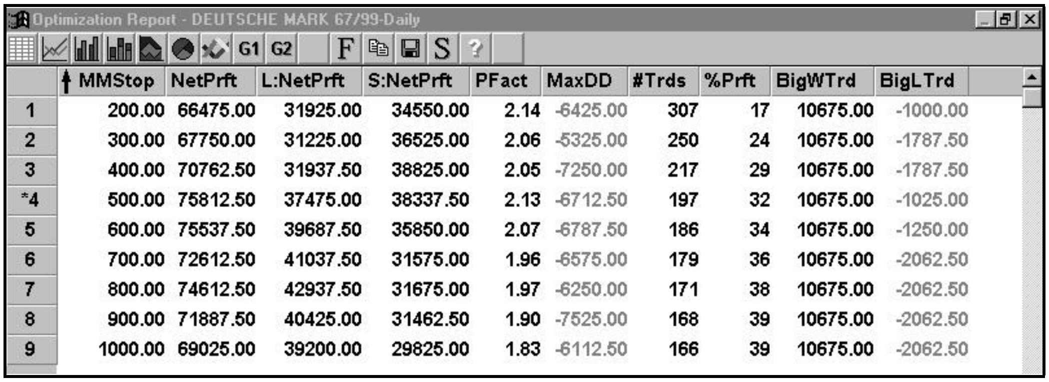


Chart 7 is the optimization report for a money-management stop. We tested the values from $200 to $1000 in increments of $100. The $500 stop gave the best results.



	MMStop	NetPrft	L:NetPrft	S:NetPrft	PFact	MaxDD	#Trds	%Prft	BigWTrd	BigLTrd
1	200.00	66475.00	31925.00	34550.00	2.14	-6425.00	307	17	10675.00	-1000.00
2	300.00	67750.00	31225.00	36525.00	2.06	-5325.00	250	24	10675.00	-1787.50
3	400.00	70762.50	31937.50	38825.00	2.05	-7250.00	217	29	10675.00	-1787.50
*4	500.00	75812.50	37475.00	38337.50	2.13	-6712.50	197	32	10675.00	-1025.00
5	600.00	75537.50	39687.50	35850.00	2.07	-6787.50	186	34	10675.00	-1250.00
6	700.00	72612.50	41037.50	31575.00	1.96	-6575.00	179	36	10675.00	-2062.50
7	800.00	74612.50	42937.50	31675.00	1.97	-6250.00	171	38	10675.00	-2062.50
8	900.00	71887.50	40425.00	31462.50	1.90	-7525.00	168	39	10675.00	-2062.50
9	1000.00	69025.00	39200.00	29825.00	1.83	-6112.50	166	39	10675.00	-2062.50

Chart 8 shows that the $500 money-management stop was the top performer in this test.

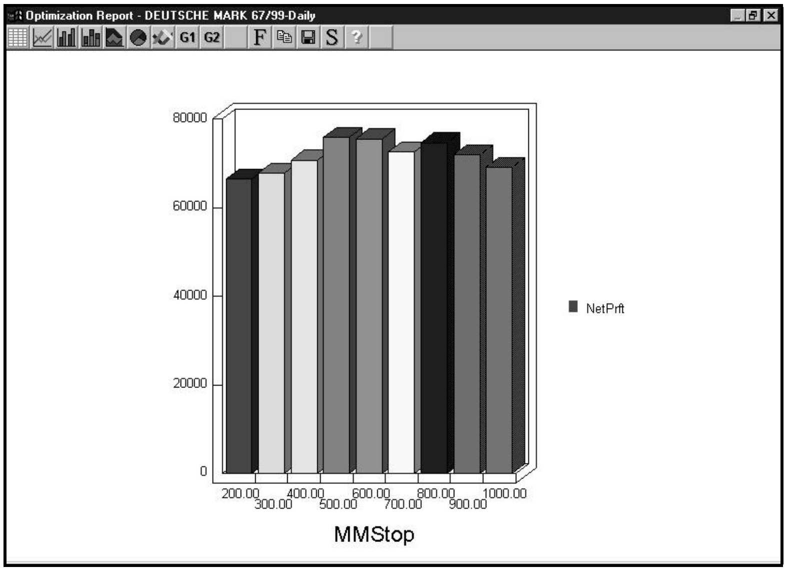
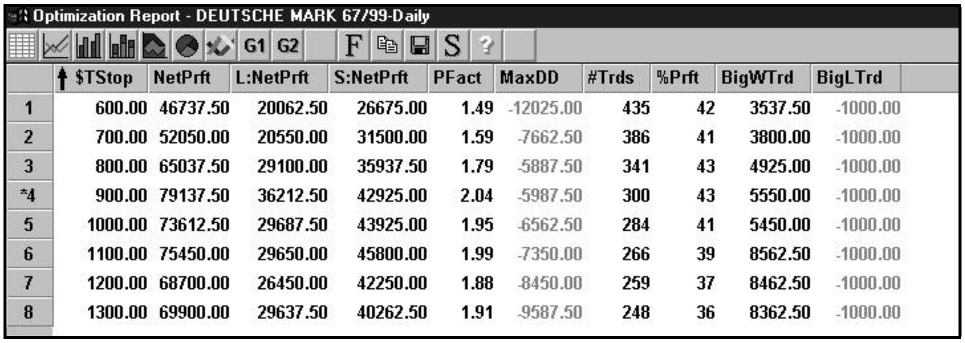


Chart 9 is the report for a $ risk trailing stop. We tested the values from $600 to $1,300 in increments of $100. The $900 risk trailing stop yielded the best results.



	$TStop	NetPrft	L:NetPrft	S:NetPrft	PFact	MaxDD	#Trds	%Prft	BigWTrd	BigLTrd
1	600.00	46737.50	20062.50	26675.00	1.49	-12025.00	435	42	3537.50	-1000.00
2	700.00	52050.00	20550.00	31500.00	1.59	-7662.50	386	41	3800.00	-1000.00
3	800.00	65037.50	29100.00	35937.50	1.79	-5887.50	341	43	4925.00	-1000.00
*4	900.00	79137.50	36212.50	42925.00	2.04	-5987.50	300	43	5550.00	-1000.00
5	1000.00	73612.50	29687.50	43925.00	1.95	-6562.50	284	41	5450.00	-1000.00
6	1100.00	75450.00	29650.00	45800.00	1.99	-7350.00	266	39	8562.50	-1000.00
7	1200.00	68700.00	26450.00	42250.00	1.88	-8450.00	259	37	8462.50	-1000.00
8	1300.00	69900.00	29637.50	40262.50	1.91	-9587.50	248	36	8362.50	-1000.00

Chart 10 shows that the $900 stop was the best of several good values.

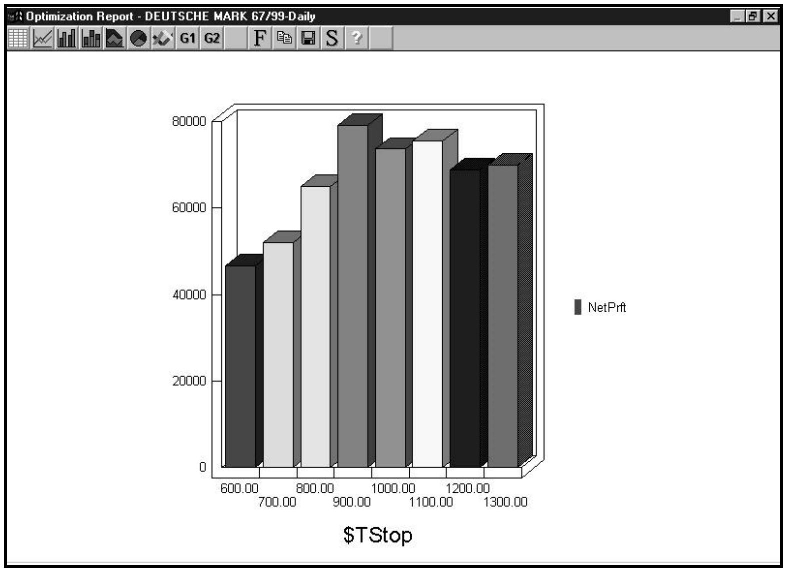
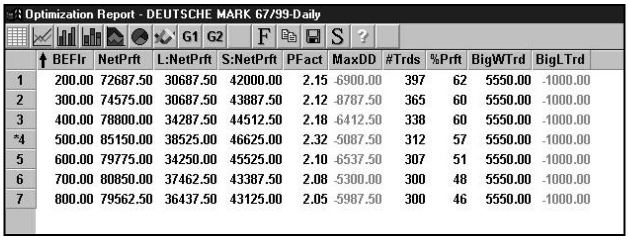


Chart 11 is the optimization report for a breakeven floor. We want to move our stop to breakeven after a trade has generated some open profits, but how many dollars in open profit should we require before we move our stop? We tested the values from $200 to $800 in increments of $100. The $500 breakeven floor proved best.



	↑ BEFlr	NetPrft	L:NetPrft	S:NetPrft	PFact	MaxDD	#Trds	%Prft	BigWTrd	BigLTrd
1	200.00	72687.50	30687.50	42000.00	2.15	-6900.00	397	62	5550.00	-1000.00
2	300.00	74575.00	30687.50	43887.50	2.12	-8787.50	365	60	5550.00	-1000.00
3	400.00	78800.00	34287.50	44512.50	2.18	-6412.50	338	60	5550.00	-1000.00
*4	500.00	85150.00	38525.00	46625.00	2.32	-5087.50	312	57	5550.00	-1000.00
5	600.00	79775.00	34250.00	45525.00	2.10	-6537.50	307	51	5550.00	-1000.00
6	700.00	80850.00	37462.50	43387.50	2.08	-5300.00	300	48	5550.00	-1000.00
7	800.00	79562.50	36437.50	43125.00	2.05	-5987.50	300	46	5550.00	-1000.00

Chart 12, a bar graph of our optimization for a breakeven stop, shows that profitability increased as the breakeven floor increased from $200 to $500 and tapered off after $500.

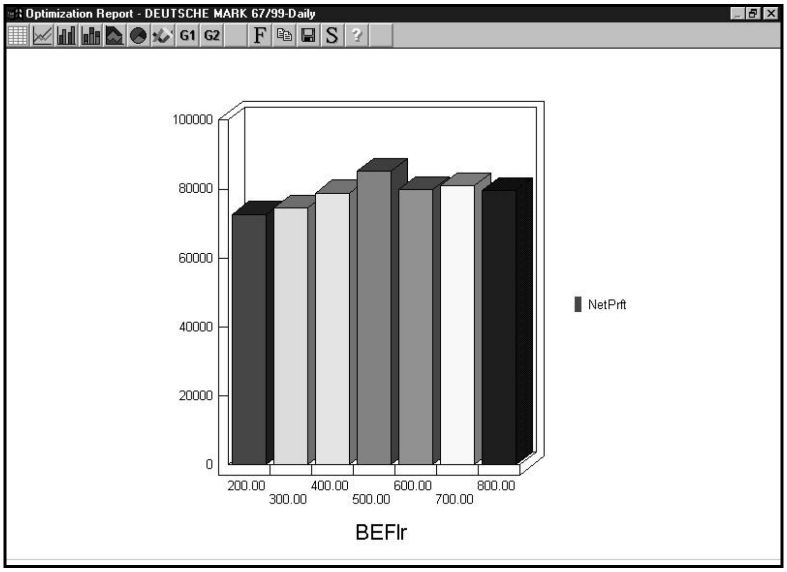


Chart 13 is the equity curve of our system now that we've added a money-management stop, a $ risk trailing stop, and a breakeven stop to our original system.

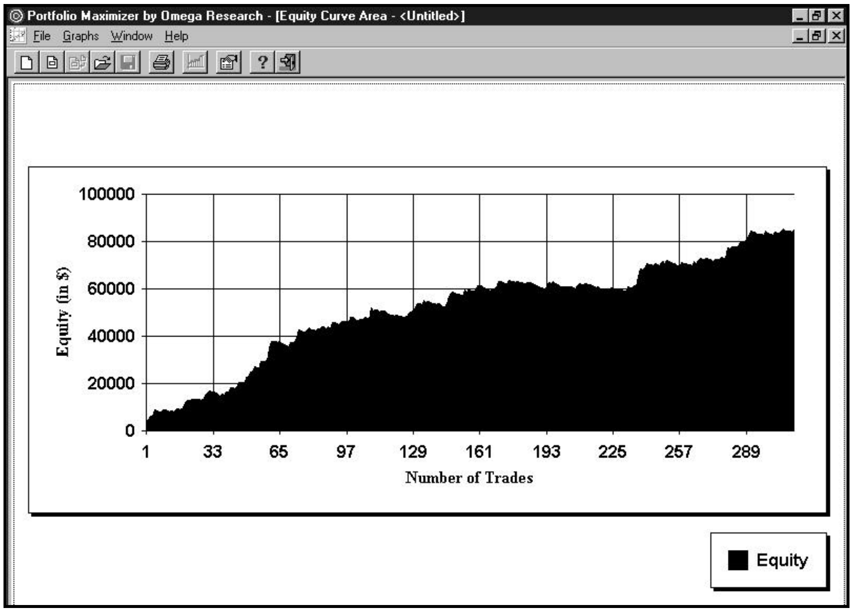


Chart 14 shows the results of our efforts in building a complete trading system from a simple idea.

Our next step, of course, would be to backward and forward test the system as it now stands on out-of-sample data. If the backward and forward tests were favorable, we could consider adding this system to the portfolio of systems we trade.

Chart 14: Ten-Bar Channel Breakout System

	STOP & REVERSE	$500 MoneyMgmt Stop	$900 Trailing Stop	$500 Breakeven Floor
Net	$72,762	$75,812	$79,137	$85,150
-$75 Slippage & Commission	$59,850	$60,462	$56,062	$61,175
# of Trades	148	197	300	312
% Wins	45	32	43	57
Profit Factor	1.88	2.13	2.04	2.32
Average Trade	491	384	263	272
Run-up	13,712	13,712	6,262	6,262
Drawdown	3,912	1,637	1,687	1,687

CHAPTER 2:

Trending Systems

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 four 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

Accumulation Swing Index System

In his 1978 classic, *New Concepts in Technical Trading Systems*, Welles Wilder introduced the Swing Index. The Swing Index looks at the relationships among the open, high, low, and close over a two-bar period. Wilder included four cross-bar comparisons and one intra-bar comparison in the Swing Index in an attempt to discover the real strength and direction of the market. The five comparisons are:

1. Close today above or below previous close

2. Close today above or below open today

3. High today above or below previous close

4. Low today above or below previous close

5. Previous close minus previous open

The Swing Index returns a number between plus 100 and minus 100 to identify short-term swing points.

The Accumulation Swing Index (ASI) indicator is a cumulative total of the Swing Index. If the Swing Index for a price bar is positive, the value is added to the ASI total. If the Swing Index for a price bar is negative, the value is subtracted from the ASI. If ASI is positive, the long-term trend is up. If it's negative, the long-term trend is down.

ASI can also be used as a breakout indicator. It signals a bullish breakout when the value of ASI exceeds the ASI value on the price bar when a swing high was established. (A swing high is a price-bar high with lower highs to its left and right.) ASI signals a bearish breakout when its value is less than it was when a swing low was established. (A swing low is a price-bar low with higher lows to its left and right.)

We wanted to base a system on an indicator that took into account more than just the closing price of each bar. Since ASI includes the opens, highs, lows, and closes of a two-bar period, it seemed like a good candidate for the main indicator of a trading system. As you will see, the results are encouraging.

Our Accumulation Swing Index System begins by applying the ASI indicator to a price chart. Then, we locate the most recent swing high or swing low with a strength of at least two. (A swing high with a strength of two has at least two consecutive lower highs to its left and right; a swing low with a strength of two has at least two consecutive higher lows to its left and right.) A setup to buy occurs when ASI is greater than it was on the most recent swing high (strength two); a setup to sell short occurs when ASI is less than it was on the most recent swing low (strength two).

After identifying a setup, we turn our attention to the entry technique. In the case of a buy setup, we enter long on a penetration of the two-bar high. When we have a sell setup, we enter short on a penetration of the two-bar low. The purpose of the entry condition is to get confirmation of the setup. We have generally achieved better results by waiting for an entry condition to be triggered than by taking a new position immediately when a setup develops.

Once we're in a new long position, we place a protective stop one point below the two-bar low. For a new short position, we set our protective stop one point above the two-bar high. We've found that our best trades move in our favor from the beginning. Few outstanding trades move very far against us, so if the trade isn't going our way early, we try to exit with only a small loss.

When long, we'll trail a stop at the most recent swing low (strength one), and when short, we'll trail a stop at the most recent swing high (strength one) to lock in profits. We'll also enable money-management, breakeven, and $ risk trailing stops.

Defining Our Trading Rules

In this system, we defined both long and short entries, although when we applied this system to a weekly stock chart (Sears), we tested the long side only. (Very few of our STAD Club members have been selling stock short on a regular basis over the last several years.) We also defined long and short exits and enabled a money-management stop, a breakeven stop, and a $ risk trailing stop. Our setup work included calculating the ASI indicator and locating the swing highs and swing lows. The setup, entries, and exits are described next.

Setup

a) Calculate the ASI indicator.

b) Locate the most recent swing high or swing low.

c) Compare the current value of ASI to its value at the most recent swing high (in the case of a positive ASI) or swing low (in the case of a negative ASI).

Long Entries

a) After a buy setup, we'll enter long at the two-bar high plus one point.

b) The buy setup will remain in effect for five bars after the setup bar.

Short Entries

a) After a sell setup, we'll enter short at the two-bar low minus one point.

b) The sell setup will remain in effect for five bars after the setup bar.

Exit Orders

a) Our initial stop when long will be one point below the two-bar low; our initial stop when short will be one point above the two-bar high.

b) When long, we'll trail a stop at the most recent swing low (strength one); when short, we'll trail a stop at the most recent swing high (strength one).

c) We'll also enable a money-management stop, a breakeven stop, and a $ risk trailing stop.

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: Accumulation Swing Index (STAD6: Acc Swing Idx)

```
Inputs: Strength(2), SwingLen(40), SetUpLen(5);
Vars: ASI(0), SH(0), SHAgo(0), SL(0), SLAgo(0), BuySetup(SetupLen), SellSetup(SetupLen);
Vars: BuyPrice(0), SellPrice(0), ILX(0), LExit(0), ISX(0), SExit(0), SHX(0), SLX(0);

{Assignment of Accumulation Swing Index and Swing values to variables}
ASI = AccumSwingIndex;
```

```
SH = SwingHighBar(1, High, Strength, Strength+1);
SHAgo = SwingHighBar(2, High, Strength, SwingLen);
SL = SwingLowBar(1, Low, Strength, Strength+1);
SLAgo = SwingLowBar(2, Low, Strength, Swinglen);
SHX = SwingHigh(1, High, 1, 2);
SLX = SwingLow(1, Low, 1, 2);
{Assignment of Stop values}
IF SHX <> -1 Then
        SExit = SHX;
IF SLX <> -1 Then
        LExit = SLX;

{Entry/Exit Setup}
IF SH <> -1 AND ASI[SH] > ASI[SHAgo] AND MarketPosition <> 1 Then Begin
        BuySetup = 0;
        BuyPrice = Highest(High, 2);
        ILX = Lowest(Low, 2);
End;
IF SL <> -1 AND ASI[SL] < ASI[SLAgo] AND MarketPosition <> -1 Then Begin
        SellSetup = 0;
        SellPrice = Lowest(Low, 2);
        ISX = Highest(High, 2);
End;

{Accumulation of bars for Setup length}
BuySetup = BuySetup + 1;
SellSetup = SellSetup + 1;

{Entries & Initial Exit}
IF BuySetup <= SetUpLen Then Begin
        Buy Next Bar at BuyPrice + 1 Point Stop;
        ExitLong Next Bar at ILX - 1 Point Stop;
End;
IF SellSetup <= SetUpLen Then Begin
        Sell Next Bar at SellPrice - 1 Point Stop;
        ExitShort Next bar at ISX + 1 Point Stop;
End;

IF MarketPosition = 1 Then
        BuySetup = SetUpLen;
IF MarketPosition = -1 Then
        SellSetup = SetUplen;

{Trailing Stops}
IF MarketPosition <> 0 Then Begin
        ExitLong Next Bar at LExit - 1 Point Stop;
        ExitShort Next Bar at SExit + 1 Point Stop;
End;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
Strength	2	Number of bars required on each side of the swing bar.
SwingLen	40	Length, expressed in bars, within which the two swings must occur.
SetUpLen	5	Length, expressed in bars, the number of bars for which the setup is valid.

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

```
Vars: ASI(0), SH(0), SHAgo(0), SL(0), SLAgo(0), BuySetup(SetupLen), SellSetup(SetupLen);
Vars: BuyPrice(0), SellPrice(0), ILX(0), LExit(0), ISX(0), SExit(0), SHX(0), SLX(0);
```

The variables above were listed in two separate variable declaration statements strictly for readability. They have the same meaning whether they are grouped together or separated as shown above.

Setup

The Accumulation Swing Index is calculated and assigned to the variable ASI. Since we are interested in knowing the ASI value on the bar of the two most recent Swings, we use the SwingHighBar and SwingLowBar Functions to identify the most recent swings. SH and SL return the number of bars since the most recent swing High/Low. SHAgo and SLAgo return the number of bars since the second most recent swing High/Low. SHX and SLX use the SwingHigh and SwingLow Functions to return the value of the most recent swing with a strength of 1. These two values are used for the Trailing Stops.

```
ASI = AccumSwingIndex;
SH = SwingHighBar(1, High, Strength, Strength+1);
SHAgo = SwingHighBar(2, High, Strength, SwingLen);
SL = SwingLowBar(1, Low, Strength, Strength+1);
SLAgo = SwingLowBar(2, Low, Strength, Swinglen);
SHX = SwingHigh(1, High, 1, 2);
SLX = SwingLow(1, Low, 1, 2);
```

The variables SExit and LExit are used to store the order price value that will be used by the Trailing Stops. To make certain that we always have a valid price value for our Exits, SExit and LExit are only assigned a value when the SHX and SLX respectively are not equal to -1. For example, when a Swing High occurs, SHX will have a value greater than -1, thus SExit will be assigned that swing value. If there is not a swing, SHX will have a value of exactly -1, so SExit will retain the value of the most recent Swing High.

```
IF SHX <> -1 Then
        SExit = SHX;
IF SLX <> -1 Then
        LExit = SLX;
```

If there is an occurrence of a Swing High (since SH is not equal to -1), and the Accumulation Swing Index on the most recent Swing High bar is greater than the Accumulation Swing Index on the second most recent Swing High bar, and the current market position is not long, then we begin a Buy Setup. The BuySetup variable, which counts the number of bars in the setup, is reset to 0. The Long Entry price is calculated, based on the highest High of the last 2 bars, and assigned to the variable BuyPrice. The initial Stop price is calculated, based on the lowest Low of the last 2 bars, and assigned to the variable ILX.

```
IF SH <> -1 AND ASI[SH] > ASI[SHAgo] AND MarketPosition <> 1 Then Begin
        BuySetup = 0;
        BuyPrice = Highest(High, 2);
        ILX = Lowest(Low, 2);
End;
```

If there is an occurrence of a Swing Low (since SL is not equal to -1), and the Accumulation Swing Index on the most recent Swing Low bar is less than the Accumulation Swing Index on the second most recent Swing Low bar, and the current market position is not short, then we begin a Sell Setup. The SellSetup variable, which counts the number of bars in the setup, is reset to 0. The Short Entry price is calculated, based on the lowest Low of the last 2 bars, and assigned to the variable SellPrice. The initial Stop price is calculated, based on the highest High of the last 2 bars, and assigned to the variable ISX.

```
IF SL <> -1 AND ASI[SL] < ASI[SLAgo] AND MarketPosition <> -1 Then Begin
        SellSetup = 0;
        SellPrice = Lowest(Low, 2);
        ISX = Highest(High, 2);
End;
```

BuySetup and SellSetup are incremented by 1 on each bar in order to keep count of the bars since the beginning of the setup. Remember that the counters below are reset each time there is a new setup.

```
BuySetup = BuySetup + 1;
SellSetup = SellSetup + 1;
```

Long Entry

If the value of the BuySetup is less than the SetUpLen value, meaning that Buy Setup has occurred, then the Long Entry and Initial Exit are placed. The Entry is a Stop order placed at the BuyPrice value, plus 1 point. The Initial Exit is a Stop order placed at the ILX value, minus 1 point.

```
IF BuySetup <= SetUpLen Then Begin
        Buy Next Bar at BuyPrice + 1 Point Stop;
        ExitLong Next Bar at ILX - 1 Point Stop;
End;
```

Short Entry

If the value of the SellSetup variable is less than the SetUpLen value, meaning that a Sell Setup has occurred, then the Short Entry and Initial Exit are placed. The Entry is a Stop order placed at the SellPrice value, minus 1 point. The Initial Exit is a Stop order placed at the ISX value, plus 1 point.

```
IF SellSetup <= SetUpLen Then Begin
        Sell Next Bar at SellPrice - 1 Point Stop;
        ExitShort Next bar at ISX + 1 Point Stop;
End;
```

When a Long order has been filled and the current position is Long, the MarketPosition function will return a 1 and the BuySetup will be set equal to the SetUpLen, thus nullifying the possibility of any additional entries by that specific setup. When a short order has been filled and the current position is Short, the MarketPosition function will return a -1 and the SellSetup will be set equal to the SetUpLen, thus nullifying the possibility of any additional Long Entries by that specific setup.

```
IF MarketPosition = 1 Then
      BuySetup = SetUpLen;
IF MarketPosition = -1 Then
      SellSetup = SetUplen;
```

Stops

If the MarketPosition function is not equal to 0, meaning that there is an outstanding position, a Long Exit Stop order is placed at the LExit value, minus 1 point. In addition, a Short Exit Stop order is placed at the SExit value, plus 1 point.

```
IF MarketPosition <> 0 Then Begin
      ExitLong Next Bar at LExit - 1 Point Stop;
      ExitShort Next Bar at SExit + 1 Point Stop;
End;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a breakeven stop and a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

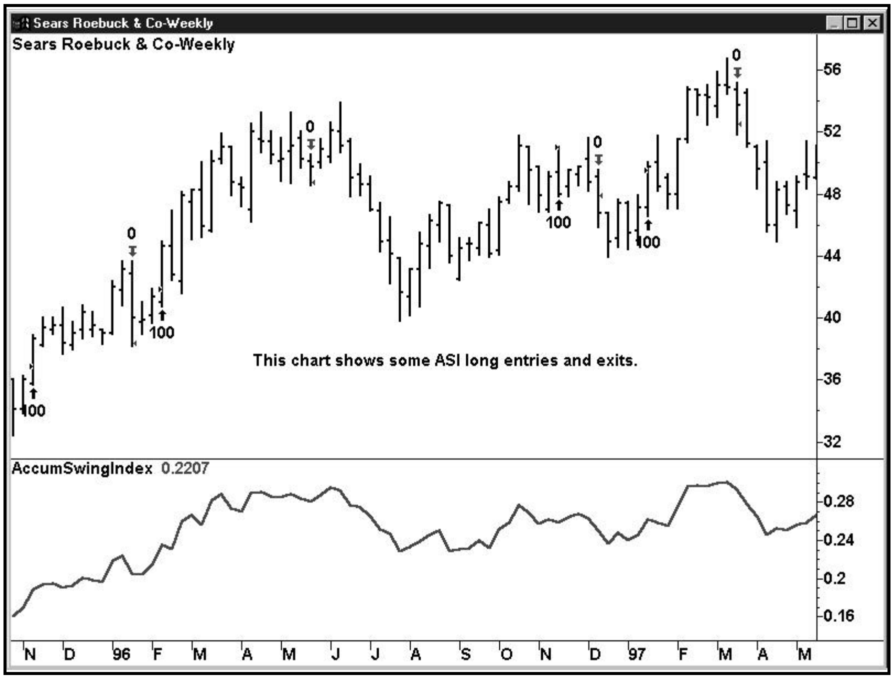
In the Properties tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a position and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We applied our ASI system to a weekly chart of Sears Roebuck & Co. from 01/78 to 10/98. [Figure 1 - weekly chart] The first Performance Summary shows the results without enabling any stops. [Figure 2 - Performance Summary] (We'll enable the stops for the next test.) The system made $2,173 on 23 trades with 57% profitable. The largest winning trade of $798.10 was not too large a percentage of the total profit. The largest losing trade, $450, wasn't too bad considering that no stops were enabled. The ratio of average win (at +$277.74) to average loss (at -$143.75) was 1.93, which is respectable when combined with 57% winning trades. The system generated four consecutive winners versus three consecutive losers. Averaging 15 bars in winning trades compared to only four bars in losing trades indicates that the system let profits run but cut losses short. [Figure 3 - Equity Curve graph]

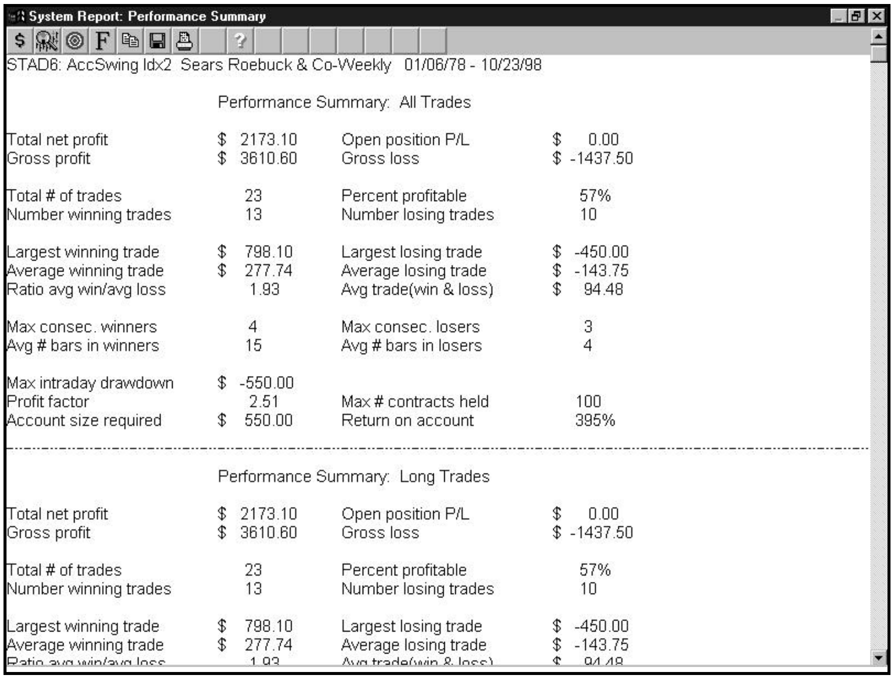
Next, we tested our ASI system with money-management, breakeven, and $ risk trailing stops enabled. [Figure 4 - weekly chart #2] Profits increased modestly from $2,173 to $2,223. Adding stops increased the number of trades from 23 to 33. Winning percent improved dramatically from 57% to 73%. The largest winning trade stayed the same at +$798.10, but the largest losing trade was cut from -$450 to -$312.50. [Figure 5 -Performance Summary #2]

We were a little disappointed to see that the maximum intraday drawdown increased from -$550 without stops to -$862.50 with stops. Enabling reasonable stops often decreases, rather than increases, drawdown.



This chart shows some ASI long entries and exits.

Figure 1 - Sears Weekly Chart



```
STAD6: AccSwing Idx2  Sears Roebuck & Co-Weekly  01/06/78 - 10/23/98

                          Performance Summary:  All Trades

Total net profit          $  2173.10    Open position P/L         $      0.00
Gross profit              $  3610.60    Gross loss                $  -1437.50

Total # of trades               23      Percent profitable              57%
Number winning trades           13      Number losing trades            10

Largest winning trade     $   798.10    Largest losing trade      $   -450.00
Average winning trade     $   277.74    Average losing trade      $   -143.75
Ratio avg win/avg loss          1.93    Avg trade(win & loss)     $     94.48

Max consec. winners              4      Max consec. losers               3
Avg # bars in winners           15      Avg # bars in losers             4

Max intraday drawdown     $  -550.00
Profit factor                   2.51    Max # contracts held           100
Account size required     $   550.00    Return on account              395%
--------------------------------------------------------------------------------
                          Performance Summary:  Long Trades

Total net profit          $  2173.10    Open position P/L         $      0.00
Gross profit              $  3610.60    Gross loss                $  -1437.50

Total # of trades               23      Percent profitable              57%
Number winning trades           13      Number losing trades            10

Largest winning trade     $   798.10    Largest losing trade      $   -450.00
Average winning trade     $   277.74    Average losing trade      $   -143.75
Ratio avg win/avg loss          1.93    Avg trade(win & loss)     $     94.48
```

Figure 2- Sears Performance Summary

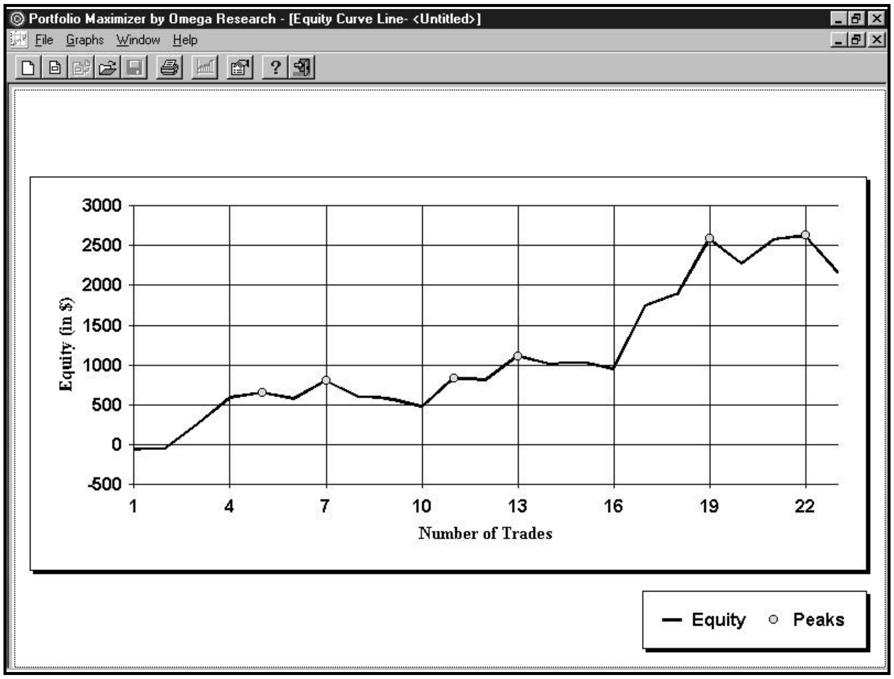
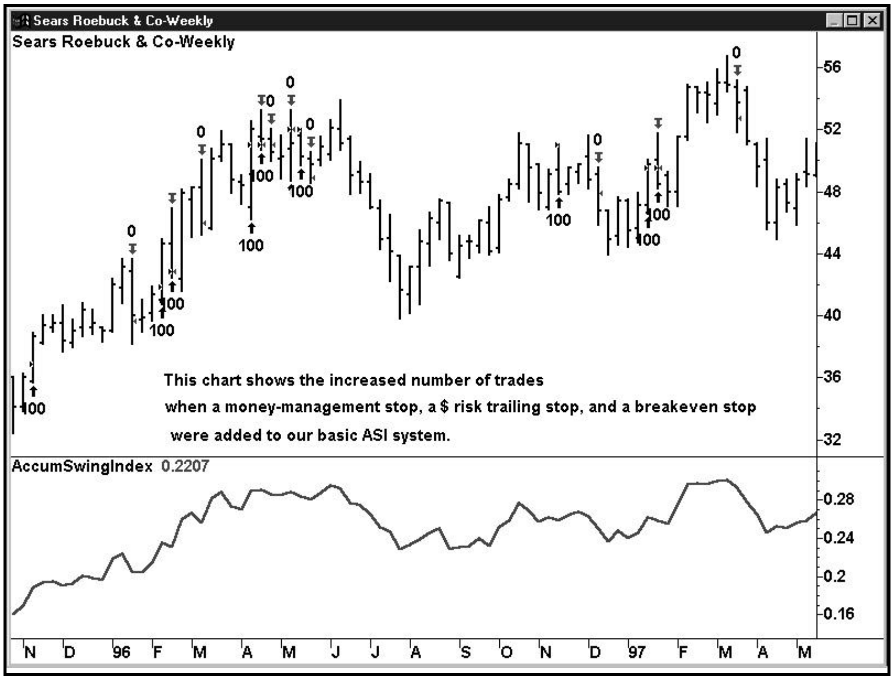
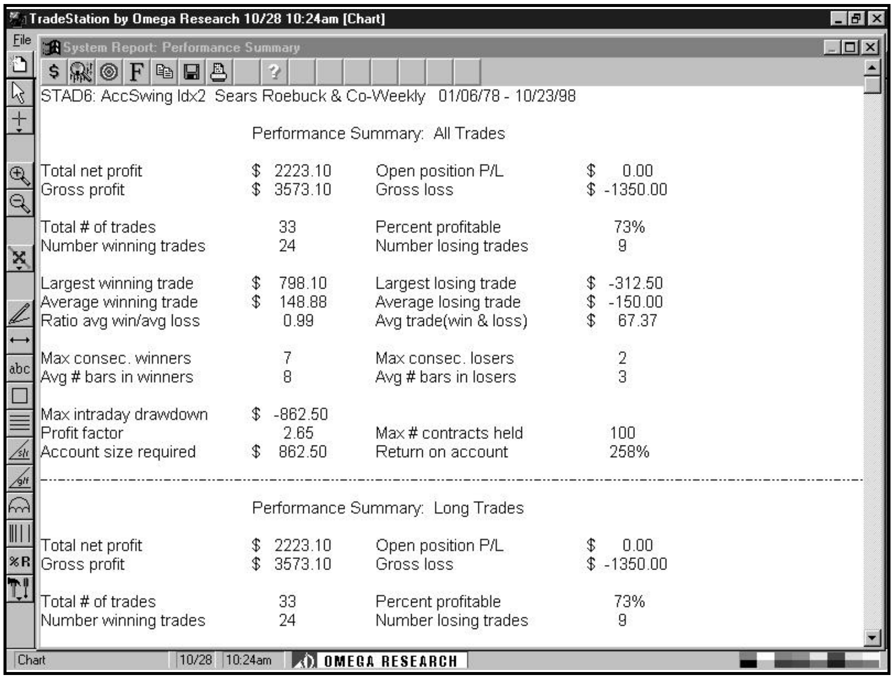


Figure 3 - Sears Equity Curve Graph



This chart shows the increased number of trades when a money-management stop, a $ risk trailing stop, and a breakeven stop were added to our basic ASI system.

Figure 4 - Sears Weekly Chart #2


TradeStation by Omega Research 10/28 10:24am [Chart]
File
System Report: Performance Summary
Chart
10/28
10:24am
OMEGA RESEARCH

```
STAD6: AccSwing Idx2  Sears Roebuck & Co-Weekly  01/06/78 - 10/23/98

                         Performance Summary:  All Trades

Total net profit         $  2223.10      Open position P/L          $     0.00
Gross profit             $  3573.10      Gross loss                 $ -1350.00

Total # of trades             33         Percent profitable              73%
Number winning trades         24         Number losing trades             9

Largest winning trade    $   798.10      Largest losing trade       $  -312.50
Average winning trade    $   148.88      Average losing trade       $  -150.00
Ratio avg win/avg loss         0.99      Avg trade(win & loss)      $    67.37

Max consec. winners            7         Max consec. losers               2
Avg # bars in winners          8         Avg # bars in losers             3

Max intraday drawdown    $  -862.50
Profit factor                  2.65      Max # contracts held           100
Account size required    $   862.50      Return on account              258%
--------------------------------------------------------------------------------

                         Performance Summary:  Long Trades

Total net profit         $  2223.10      Open position P/L          $     0.00
Gross profit             $  3573.10      Gross loss                 $ -1350.00

Total # of trades             33         Percent profitable              73%
Number winning trades         24         Number losing trades             9
```

Figure 5 - Sears Performance Summary #2

Suggestions For Improvement

Although our ASI system is a trend-follower, it did not generate buy signals for all the substantial uptrends in the 20-year period we tested. The system could probably be improved by adding a fail-safe entry so that we would participate in all the good up-moves. An example of a fail-safe entry technique would be to buy any new 40-bar high even if the ASI system had not signaled a long entry.

ADX & Moving-Average Channel System

ADX (Average Directional Index) and moving averages are the two most widely used technical indicators for trend-followers. We thought that a system combining the two indicators would probably yield good results, and we weren't disappointed.

ADX measures the amount of consistent directional movement ("trendiness") in a market. When ADX is rising, a market is considered to be trending; when it's falling, a market is considered not trending (moving in a trading range).

An exponential moving average (EMA) smooches price data to make the underlying trend easier to discern. Many technicians prefer EMAs to simple moving averages because EMAs give more weight to the most recent data and don't abruptly drop older data when it falls out of the lookback period. A ten-bar EMA, for example, would count the current closing price as 18% of the new EMA (as opposed to the 10% it would count if all closing prices were weighted equally). The EMA up to the most recent close would count 82% of the new EMA, thus including all the closing prices in the data series under consideration.

For this system, we calculated an EMA of highs and an EMA of lows to construct a moving-average channel (MAC). Our setup to buy is a close above the MAC with ADX greater than it was one bar ago; our setup to sell short is a close below the MAC with ADX greater than one bar ago. (Remember that a rising ADX indicates a strong trend, but the trend can be either an uptrend or a downtrend.)

With the setup in place, we need to calculate our entry price. With a close above the MAC and ADX rising, we will enter long at the close of the setup bar plus half the height of the MAC (EMA of highs minus EMA of lows)/2. If an entry is not triggered within two bars of the setup, the setup is cancelled. Our exit when long will be on a close below the MAC.

When a market closes below the MAC with ADX rising, we have a setup to sell short. Our entry price will be at the close of the setup bar minus half the height of the MAC. We cancel the setup if the entry is not triggered within two bars.

Defining Our Trading Rules

In this system, we defined both long and short entries as well as exit orders. We also did some setup work to construct the EMA channel of highs and lows and to calculate the ADX. The setup, entries, and exits are described next.

Setup

a) Calculate a 30-bar EMA of highs and a 30-bar EMA of lows.

b) Calculate a 12-bar ADX.

Long Entries

a) Check for a close above the EMA of highs and a rising ADX.

b) If both conditions are true on the same bar, calculate the entry price by adding half the height of the MAC to the close of the setup bar.

c) Keep the entry order active for two bars after the setup.

Short Entries

a) Check for a close below the EMA of lows and a rising ADX.

b) If both conditions are true on the same bar, calculate the entry price by subtracting half the height of the MAC from the close of the setup bar.

c) Keep the entry order active for two bars after the setup.

Exit Orders

a) Exit a long position on a close below the EMA of lows.

b) Exit a short position on a close above the EMA of highs.

c) Also enable a money management stop, a breakeven stop, and a $ risk trailing stop.

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: Moving Average Channel & ADX (STAD6: MovAvg & ADX)

```
Inputs: AvgLen(30), ADXLen(12), EntryBar(2);
Vars: UpperMA(0), LowerMA(0), ADXValue(0), ChanSpread(0), BuySetup(False), SellSetup(False),
BuyTarget(0), SellTarget(0), MROBS(0), MROSS(0);

{Variable Declarations}
UpperMA = XAverage(High, AvgLen)[1];
LowerMA = XAverage(Low, AvgLen)[1];
ADXValue = ADX(ADXLen);
ChanSpread = (UpperMA - LowerMA) / 2;
```

```
{Setup}
BuySetup = Close > UpperMA AND ADXValue > ADXValue[1];
SellSetup = Close < LowerMA AND ADXValue > ADXValue[1];

IF BuySetup Then
        BuyTarget = Close + ChanSpread;
IF SellSetup Then
        SellTarget = Close - ChanSpread;

MROBS = MRO(BuySetup, EntryBar, 1);
MROSS = MRO(SellSetup, EntryBar, 1);

{Entries}
IF MROBS <> -1 AND MRO(MarketPosition=1, MROBS, 1) = -1 Then
        Buy Next Bar at BuyTarget Stop;
IF MROSS <> -1 AND MRO(MarketPosition=-1, MROSS, 1) = -1 Then
        Sell Next Bar at SellTarget Stop;

{Exits}
ExitLong Next Bar at UpperMA - 1 Point Stop;
ExitShort Next Bar at LowerMA + 1 Point Stop;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
AvgLen	30	Length, expressed in bars, used to calculate the exponential moving average of the High and Low.
ADXLen	12	Length, expressed in bars, used to calculate the Average Directional Index (ADX).
EntryBar	2	Length, expressed in bars, for which the entry setup is valid.

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

```
Vars: UpperMA(0), LowerMA(0), ADXValue(0), ChanSpread(0), BuySetup(False), SellSetup(False),
BuyTarget(0), SellTarget(0), MROBS(0), MROSS(0);
```

Setup

Initially, we define the Upper Moving Average Band (UpperMA), and the Lower Moving Average Band (LowerMA) using an Exponential Moving Average. The value of the ADX is assigned to the variable ADXVal. Finally, the value of half the High/Low channel range is assigned to the variable ChanSpread.

```
UpperMA = XAverage(High, AvgLen)[1];
LowerMA = XAverage(Low, AvgLen)[1];
ADXValue = ADX(ADXLen);
ChanSpread = (UpperMA - LowerMA) / 2;
```

BuySetup and SellSetup are True/False variables that determine when a Buy/Sell setup has occurred. For a BuySetup to occur, the Close must be above the Upper Moving Average band, and the ADX value must be greater than the ADX value on the previous bar. For a SellSetup to occur, the Close must be below the Lower Moving Average band, and the ADX value must be lower than the ADX value on the previous bar.

```
BuySetup = Close > UpperMA AND ADXValue > ADXValue[1];
SellSetup = Close < LowerMA AND ADXValue > ADXValue[1];
```

If a BuySetup occurs, a BuyTarget value for the Long Entry is defined by adding the ChanSpread value to the Close. If a SellSetup occurs, a SellTarget value for the Short Entry is defined by subtracting the ChanSpread value from the Close.

```
IF BuySetup Then
       BuyTarget = Close + ChanSpread;
IF SellSetup Then
       SellTarget = Close - ChanSpread;
```

The MROBS variable uses the MRO function to determine if a valid BuySetup has occurred within the last EntryBar bars. If a BuySetup has occurred, the function will return a value greater or equal to 0, otherwise it will return a value of -1. The MROSS also uses the MRO function to determine if a valid SellSetup has occurred within the last EntryBar bars. If a SellSetup has occurred, the function will return a value greater or equal to 0, otherwise it will return a value of -1.

```
MROBS = MRO(BuySetup, EntryBar, 1);
MROSS = MRO(SellSetup, EntryBar, 1);
```

Long Entry

There are two criteria required for the placement of a Buy order. First, the MROBS variable must not be equal to -1, which indicates the occurrence of a BuySetup. The second criterion uses the MRO and MarketPosition functions to determine if there has been a long entry since the confirmation of a valid BuySetup. The MRO function must return a -1 (meaning that there has not been a Long Entry since the occurrence of the BuySetup) in order for the criterion to be true. This insures that only one Buy signal can result from a given valid BuySetup. If both criteria have been met, a Buy stop order is placed at the BuyTarget value that was calculated above.

```
IF MROBS <> -1 AND MRO(MarketPosition=1, MROBS, 1) = -1 Then
       Buy Next Bar at BuyTarget Stop;
```

Short Entry

There are also two criteria required for the placement of a Sell order. First, the MROSS variable must not be equal to -1, which indicates the occurrence of a SellSetup. The second criterion uses the MRO and MarketPosition functions to determine if there has been a short entry since the confirmation of a valid SellSetup. The MRO function must return a -1 (meaning that there has not been a Short Entry since the occurrence of the SellSetup) in order for the criterion to be true. This insures that only one Sell signal can result from a given valid SellSetup. If both criteria have been met, a Sell stop order is placed at the SellTarget value that was calculated above.

```
IF MROSS <> -1 AND MRO(MarketPosition=-1, MROSS, 1) = -1 Then
       Sell Next Bar at SellTarget Stop;
```

Exits

For a Long Exit, a Stop order is placed at 1 point below the lower moving average value. For the Short Exit, a Stop order is placed at 1 point above the higher moving average value.

```
ExitLong Next Bar at UpperMA - 1 Point Stop;
ExitShort Next Bar at LowerMA + 1 Point Stop;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a breakeven stop and a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option Do not allow multiple entries in the same direction. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We tested this system on daily Intel data from 01/88 to 10/98 (on the long side only). [Figure 6 - Intel daily chart] The system earned $5,735 (per 100 shares) on 51 trades with 51% profitable. [Figure 7 - Performance Summary] The largest winning trade ($1,675) was more than four times as big as the largest losing trade (-$400), and the average winner ($318) was more than three times the size of the average loser (-$101). The average trade made $112. Our system let profits run for an average of 33 bars and cut losses short by exiting losing trades in just ten bars. The maximum intraday drawdown (-$700) was only 12% of the total net profit ($5,735). The profit factor (3.26) indicates that the system earned $3.26 for each $1.00 it lost. Finally, the Average Profit By Month graph shows that only two months (March and November) failed to make money on average during the ten-year test period. [Figure 8 - Average Profit By Month graph]

We also tested the system on daily Japanese Yen data from 01/95 to 10/98, using the same parameters as the Intel test, but not enabling any stops. [Figure 9 - JY daily chart]

On 28 trades, the system produced $27,312 in net profit with 39% of the trades profitable. [Figure 10 -JY Performance Summary] The average trade (both wins and losses) made $975 per contract. The system stayed in winning trades an average of 32 bars but exited losers in an average of nine bars. The ratio of average win to average loss was good (2.93), but the profit factor was only fair at $1.90. Another negative was that the system suffered seven consecutive losing trades while posting only two consecutive winning trades.

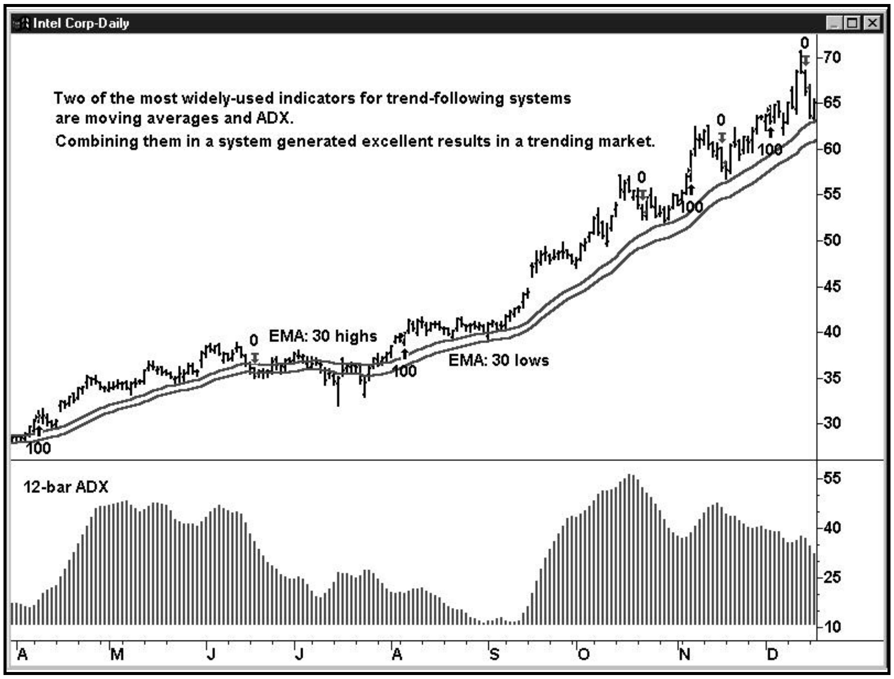
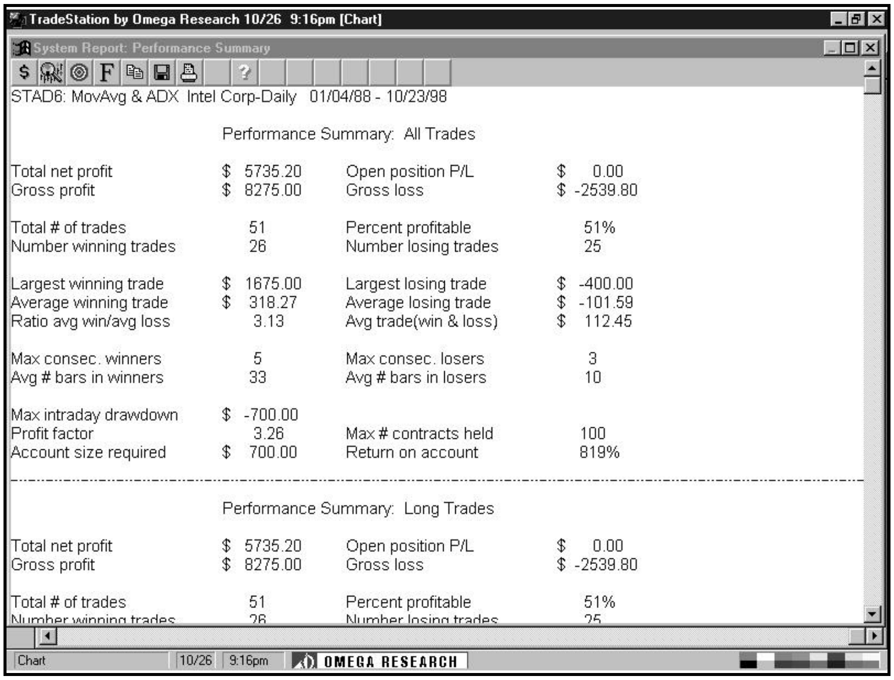

Two of the most widely-used indicators for trend-following systems are moving averages and ADX.
Combining them in a system generated excellent results in a trending market.

Figure 6 - Intel Daily Chart


TradeStation by Omega Research 10/26 9:16pm [Chart]
System Report: Performance Summary
$
F
?
Chart
10/26
9:16pm
OMEGA RESEARCH

```
STAD6: MovAvg & ADX  Intel Corp-Daily   01/04/88 - 10/23/98

                              Performance Summary:  All Trades

Total net profit              $  5735.20     Open position P/L            $     0.00
Gross profit                  $  8275.00     Gross loss                   $ -2539.80

Total # of trades                51          Percent profitable              51%
Number winning trades            26          Number losing trades            25

Largest winning trade         $  1675.00     Largest losing trade         $  -400.00
Average winning trade         $   318.27     Average losing trade         $  -101.59
Ratio avg win/avg loss             3.13      Avg trade(win & loss)        $   112.45

Max consec. winners               5          Max consec. losers               3
Avg # bars in winners            33          Avg # bars in losers            10

Max intraday drawdown         $  -700.00
Profit factor                      3.26      Max # contracts held           100
Account size required         $   700.00     Return on account              819%
-------------------------------------------------------------------------------------------

                              Performance Summary:  Long Trades

Total net profit              $  5735.20     Open position P/L            $     0.00
Gross profit                  $  8275.00     Gross loss                   $ -2539.80

Total # of trades                51          Percent profitable              51%
Number winning trades            26          Number losing trades            25
```

Figure 7 - Intel Performance Summary

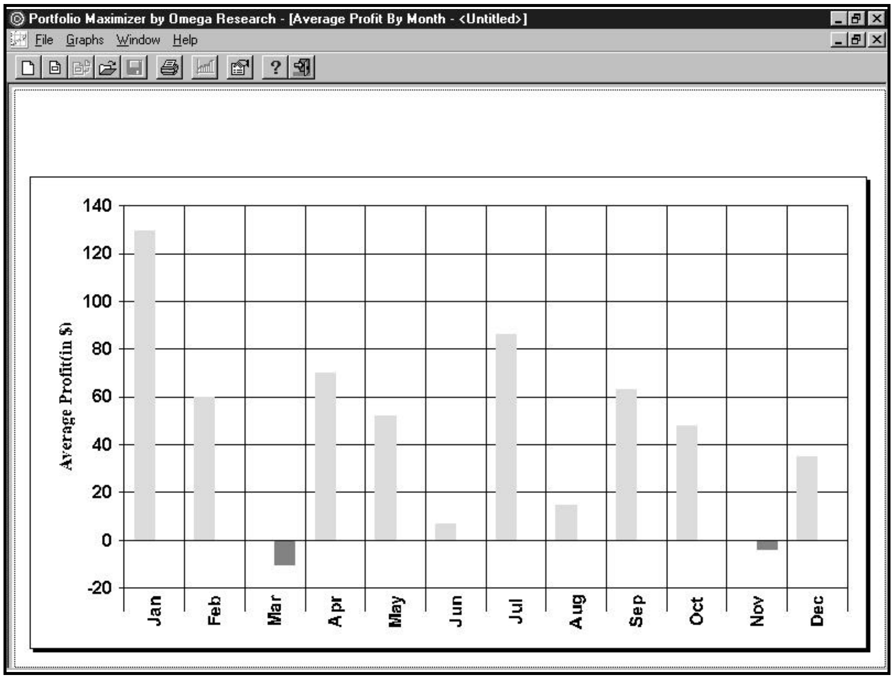


Figure 8 - Intel Average Profit By Month Graph

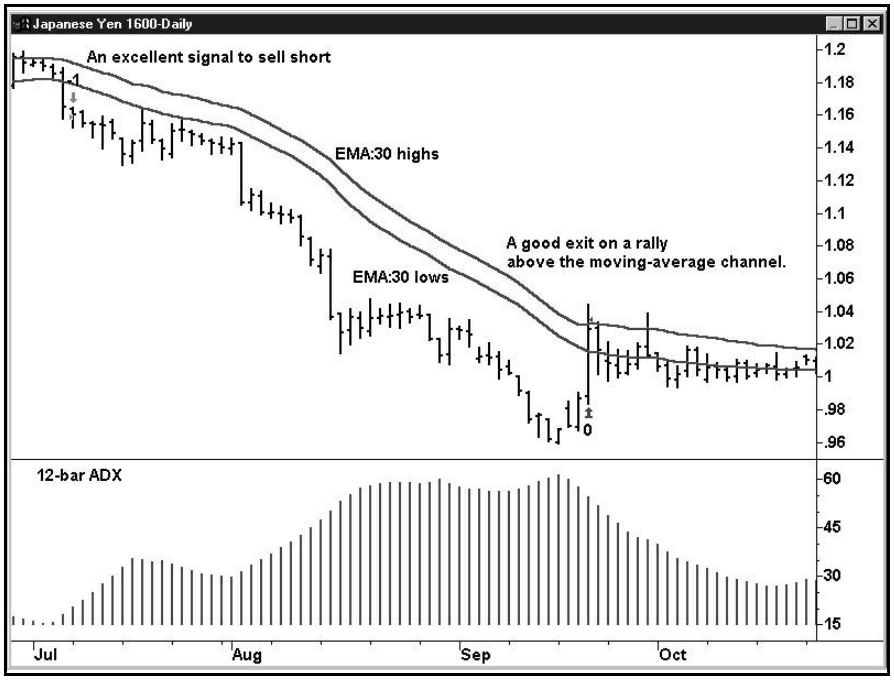
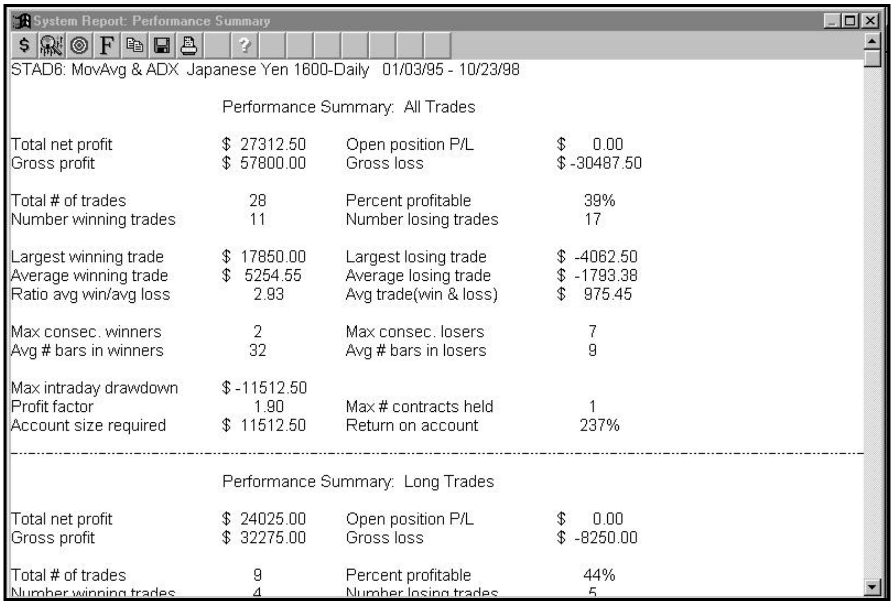


Figure 9 - Japanese Yen Daily Chart



```
STAD6: MovAvg & ADX  Japanese Yen 1600-Daily   01/03/95 - 10/23/98

                          Performance Summary:  All Trades

Total net profit          $  27312.50    Open position P/L          $     0.00
Gross profit              $  57800.00    Gross loss                 $ -30487.50

Total # of trades              28        Percent profitable             39%
Number winning trades          11        Number losing trades           17

Largest winning trade     $  17850.00    Largest losing trade       $  -4062.50
Average winning trade     $   5254.55    Average losing trade       $  -1793.38
Ratio avg win/avg loss          2.93     Avg trade(win & loss)      $   975.45

Max consec. winners             2        Max consec. losers              7
Avg # bars in winners          32        Avg # bars in losers            9

Max intraday drawdown     $ -11512.50
Profit factor                   1.90     Max # contracts held            1
Account size required     $  11512.50    Return on account             237%
--------------------------------------------------------------------------------

                          Performance Summary:  Long Trades

Total net profit          $  24025.00    Open position P/L          $     0.00
Gross profit              $  32275.00    Gross loss                 $  -8250.00

Total # of trades               9        Percent profitable             44%
Number winning trades           4        Number losing trades            5
```

Figure 10 - Japanese Yen Performance Summary

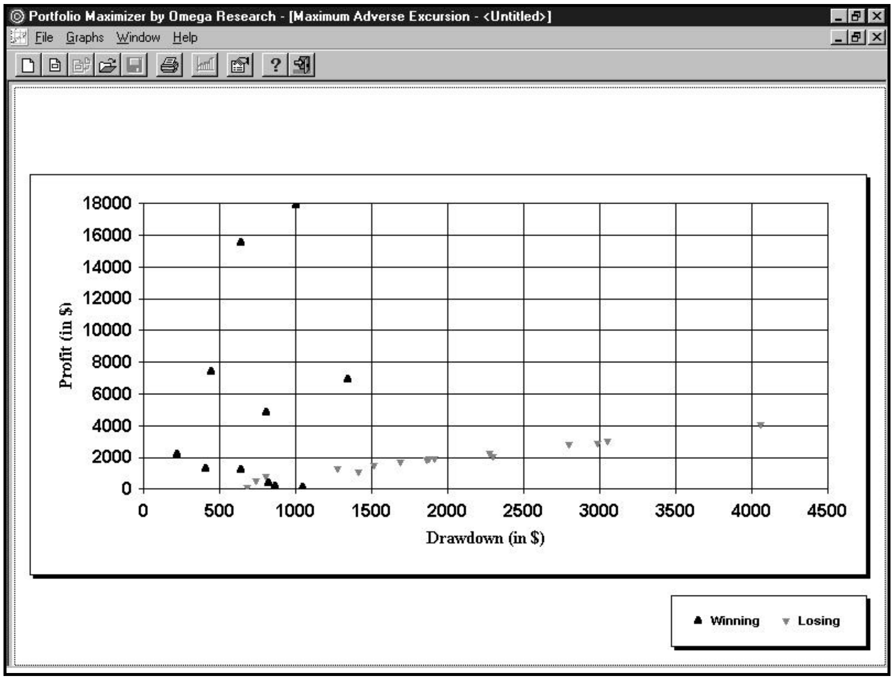


Figure 11 - Japanese Yen Maximum Excursion Adverse Excursion Graph

Suggestions For Improvement

The final graph we'll include for this system is Portfolio Maximizer's Maximum Adverse Excursion graph. [Figure 11 - Maximum Adverse Excursion graph] Perhaps it will show us a way to improve this system. The vertical axis represents profit (or loss) in dollars; the horizontal axis represents the drawdown (or maximum adverse excursion) of each trade. The upward-pointing triangles are winning trades, and the downward-pointing triangles are losing trades.

The most significant piece of information that this particular graph provides is that only one winning trade suffered a drawdown of $1,000 or more, but 12 losing trades had a drawdown in excess of $1,000. It's clear that a money-management stop of approximately $1,000 would reduce dollars lost more than it would reduce dollars won, thus improving the system. We would try to improve this system by adding a money-management stop, a breakeven stop, and either a $ risk trailing stop or a % risk trailing stop.

Displaced Moving-Average Channel & Range Leaders System

In STAD Club 4, we included a system based on a displaced moving average with volume as a confirming indicator. In STAD Club 6, we'll construct a displaced moving-average channel of highs and lows and add a simple chart pattern known as a Range Leader.

As discussed in STAD Club 4, a displaced moving average is shifted forward a specified number of bars rather than plotted on the bar for which it was calculated. This allows us to know the numeric value the moving average will have on a bar in the future. Many traders believe that displaced moving averages can reduce the "whipsaws" (the many small losses due to false signals) that occur when a standard (not displaced) moving average is applied to a price series.

For this system, we'll construct a displaced moving-average channel by calculating 20-bar moving averages of highs and lows and shifting them forward by five bars.

A bullish Range Leader is a bar with a midpoint above the previous bar's high and a range greater than the previous bar's range; a bearish Range Leader has a midpoint below the previous bar's low and a range greater than the previous bar's range.

The setup to buy is a bullish Range Leader. The entry will usually be one point above the DMA channel. However, if the close of the Range Leader is above the channel, we'll buy on the close. The setup to sell short is a bearish Range Leader. The entry will usually be one point below the DMA channel, but if the close of the Range Leader is below the channel, we'll sell short on the close.

When we're in a long position, we'll trail a stop one point below the DMA channel; when short, we'll trail a stop one point above the channel. We'll also enable a money-management stop, a breakeven stop, and a $ risk trailing stop.

Defining Our Trading Rules

In this system, we defined both long and short entries, as well as exit orders. We also did some setup work to calculate the displaced moving-average channel of highs and lows. The setup, entries, and exits are described next.

Setup

a) Calculate a 20 x 5 displaced moving average of highs and a 20 x 5 displaced moving average of lows.

Long Entries

a) Our setup to buy is a bullish Range Leader bar.

b) Our long entry is one point above the DMA channel or on the close of the Range Leader bar if the close is above the DMA channel.

Short Entries

a) Our setup to sell short is a bearish Range Leader bar.

b) Our short entry is one point below the DMA channel of on the close of the Range Leader bar if the close is below the DMA channel.

Exit Orders

a) For the bars from the entry bar through the "ExitBar" variable, an exit will be generated at a displaced moving average based on the median of each bar.

b) After the system has been long for "ExitBar" or more bars, the exit is one point above the DMA channel.

c) After the system has been short for "ExitBar" or more bars, the exit is one point above the DMA channel.

d) We'll also enable a money-management stop, a breakeven stop, and a $ risk trailing stop.

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: Displaced Moving Average Channel (STAD6: MovAvgDChan)

```
Inputs: AvgLen(20), Disp(-5), ExitBar(5);
Vars: UpperAvg(0), LowerAvg(0), ExitAvg(0), AbsDisp(0), RangeLeadB(False), RangeLeadS(False);

{Assignment of values for Displaced Moving Average Channel}
AbsDisp = AbsValue(Disp);
UpperAvg = Average(High, AvgLen)[AbsDisp];
LowerAvg = Average(Low, AvgLen)[AbsDisp];
ExitAvg = Average(MedianPrice, AvgLen)[AbsDisp];

{Variables which determine if there has been a Range Leader Bar}
RangeLeadB = MedianPrice > High[1] AND Range > Range[1];
RangeLeadS = MedianPrice < Low[1] AND Range > Range[1];

{Entries}
IF RangeLeadB AND Close > UpperAvg Then
        Buy This Bar on Close;
IF RangeLeadS AND Close < LowerAvg Then
        Sell This Bar on Close;

{Exits}
IF BarsSinceEntry < ExitBar Then Begin
        ExitLong Next Bar at ExitAvg Stop;
        ExitShort Next Bar at ExitAvg Stop;
End
Else Begin
        ExitLong Next Bar at UpperAvg - 1 Point Stop;
        ExitShort Next Bar at LowerAvg + 1 Point Stop;
End;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
AvgLen	20	Length, expressed in bars, used to calculate the Moving Average Channels.
Disp	-5	The number of bars which the Moving Average Channels will be displaced.
ExitBar	5	The number of bars for which the initial exit is valid.

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

```
Vars: UpperAvg(0), LowerAvg(0), ExitAvg(0), AbsDisp(0), RangeLeadB(False), RangeLeadS(False);
```

Setup

First we calculate the Moving Average Channels. For the purposes of this System, the channels can only be displaced forward (to the right). Thus, we use the variable AbsDisp to return the Absolute Value of the Disp Input. The displacement value is then used in the calculation of the channels. In addition to the channels, we also calculate a displaced Moving Average of the Median Price, which will be used as the initial exit.

```
AbsDisp = AbsValue(Disp);
UpperAvg = Average(High, AvgLen)[AbsDisp];
LowerAvg = Average(Low, AvgLen)[AbsDisp];
ExitAvg = Average(MedianPrice, AvgLen)[AbsDisp];
```

An important part of the setup is the identification of Range Leader Bars. RangeLeadB is used to identify a Bullish Range Leader. If the Median Price is greater than the previous bar's High and the Range is greater than the previous bar's range, RangeLeadB returns True. RangeLeadS is used to identify a Bearish Range Leader. If the MedianPrice is less than the previous bar's Low and the Range is greater than the previous bar's Range, then RangeLeadS returns True.

```
RangeLeadB = MedianPrice > High[1] AND Range > Range[1];
RangeLeadS = MedianPrice < Low[1] AND Range > Range[1];
```

Entries

When RangeLeadB is True, indicating the occurrence of a Bullish Range Leader bar, and the Close is greater than the Upper Moving Average band, a Long Entry is generated on the Close of the bar. When RangeLeadS is True, indicating the occurrence of a Bearish Range Leader bar, and the Close is less than the Lower Moving Average band, a Short Entry is generated on the Close of the bar.

```
IF RangeLeadB AND Close > UpperAvg Then
        Buy This Bar on Close;
IF RangeLeadS AND Close < LowerAvg Then
        Sell This Bar on Close;
```

Exits

The initial exits for the system are used while the number of bars since the entry bar is less than the value specified in the ExitBar input. During this period, both the Long and Short Exits are generated at the ExitAvg.

```
IF BarsSinceEntry < ExitBar Then Begin
        ExitLong Next Bar at ExitAvg Stop;
        ExitShort Next Bar at ExitAvg Stop;
End
```

Once the number of bars since the entry has exceeded the ExitBar value, a new trailing stop takes effect. The Long Exit is placed at the Upper Average Channel value, minus 1 point. The Short Exit is placed at the Lower Average Channel value, plus 1 point.

```
Else Begin
        ExitLong Next Bar at UpperAvg - 1 Point Stop;
        ExitShort Next Bar at LowerAvg + 1 Point Stop;
End;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a breakeven stop and a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We tested this system on weekly Wal-Mart data from 07/79 to 10/98. [Figures 12 & 13 - Wal-Mart weekly charts] Although the system lost $425 on the short trades, we're including both the long and the short trade results in the All Trades Performance Summary. [Figure 14 - Performance Summary]

Our system made $2,713 on 54 trades, with 57% profitable. At $2,056, the largest winning trade was almost ten times bigger than the largest losing trade of $206. The ratio of average win to average loss was 1.6, and the profit factor (helped by the favorable 57% winners) was 2.16. (Considering long trades only, the profit factor was an excellent 3.22.)

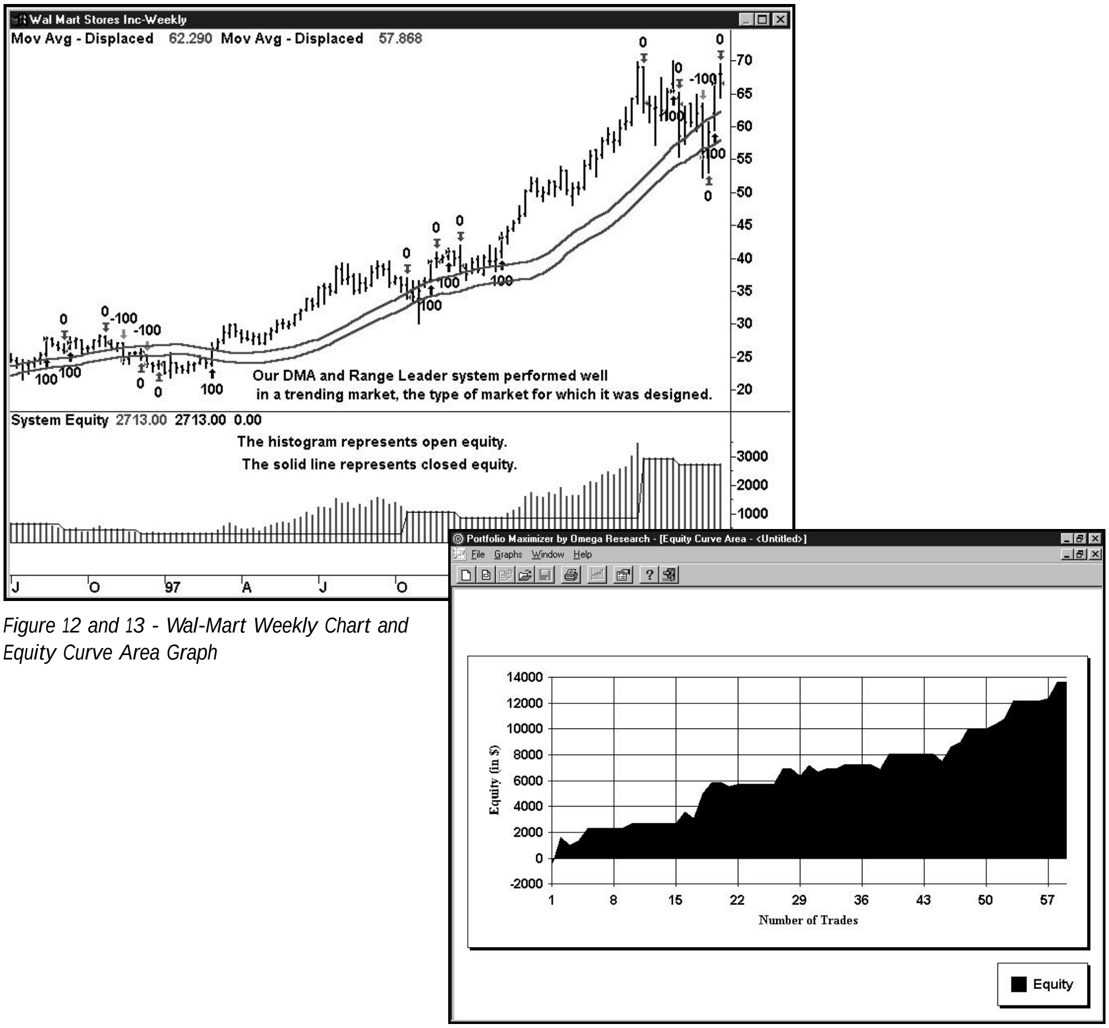
This system also did a very good job of letting profits run and cutting losses short, averaging 18 bars in the winners versus only four bars in the losers.

The Total Trades graph shows one positive outlier and no negative outliers (trades more than three standard deviations from an average trade). [Figure 15 - Total Trades graph] Although it's certainly better to have one positive outlier and no negative ones compared to the reverse, the scarcity of positive outliers explains why the system only made $50.24 per trade, even with 57% winners.

We also tested this system on British Pound daily data from 10/96 to 10/98 with the same 20 x 5 DMA Channel. [Figure 16 - BP daily chart] The breakeven stop and $ risk trailing stop remained the same (at $100 and $600, respectively). The only parameter change was from a $200 money-management stop for Wal-Mart to a $900 money-management stop for the British Pound.

The system earned $13,637 per contract in the two-year period. [Figure 17 - BP Performance Summary] Of the 59 trades, 86% were profitable. Although the largest winner ($2,087) was much greater than the largest loser (-$600), the average winner ($345) was less than the average loser (-$498). With 86% profitable trades, that's acceptable, especially considering that the average trade (win and loss) made $231. The 4.42 profit factor is exceptional. Our system earned $4.42 for each dollar it lost.

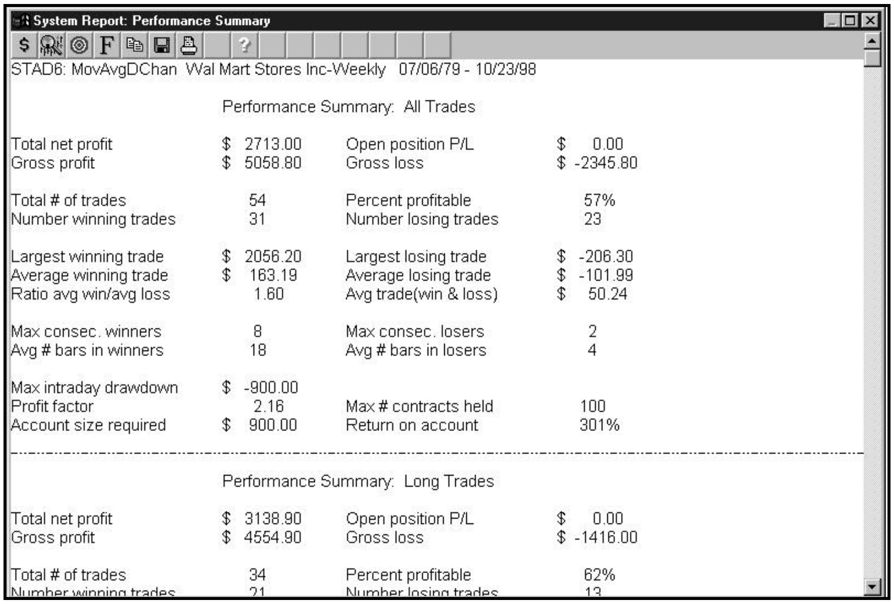
The equity curve, which was produced in Omega Research's Portfolio Maximizer, shows a steady increase in equity without any serious declines. [Figure 18 - Equity Curve graph]


Wal Mart Stores Inc-Weekly
Mov Avg - Displaced 62.290 Mov Avg - Displaced 57.868
70
65
60
55
50
45
40
35
30
25
20
100
0
-100
System Equity 2713.00 2713.00 0.00
3000
2000
1000
J
O
97
A
J
O
Portfolio Maximizer by Omega Research - [Equity Curve Area - <Untitled>]
File Graphs Window Help
14000
12000
10000
8000
6000
4000
2000
0
-2000
Equity (in $)
1
8
15
22
29
36
43
50
57
Number of Trades
■ Equity

Our DMA and Range Leader system performed well in a trending market, the type of market for which it was designed.

The histogram represents open equity.
The solid line represents closed equity.

Figure 12 and 13 - Wal-Mart Weekly Chart and Equity Curve Area Graph



```
STAD6: MovAvgDChan  Wal Mart Stores Inc-Weekly  07/06/79 - 10/23/98

                            Performance Summary:  All Trades

Total net profit          $  2713.00     Open position P/L          $     0.00
Gross profit              $  5058.80     Gross loss                 $ -2345.80

Total # of trades             54         Percent profitable             57%
Number winning trades         31         Number losing trades           23

Largest winning trade     $  2056.20     Largest losing trade       $  -206.30
Average winning trade     $   163.19     Average losing trade       $  -101.99
Ratio avg win/avg loss         1.60      Avg trade(win & loss)      $    50.24

Max consec. winners            8         Max consec. losers              2
Avg # bars in winners         18         Avg # bars in losers            4

Max intraday drawdown     $  -900.00
Profit factor                  2.16      Max # contracts held          100
Account size required     $   900.00     Return on account             301%
------------------------------------------------------------------------------

                            Performance Summary:  Long Trades

Total net profit          $  3138.90     Open position P/L          $     0.00
Gross profit              $  4554.90     Gross loss                 $ -1416.00

Total # of trades             34         Percent profitable             62%
Number winning trades         21         Number losing trades           13
```

Figure 14 - Wal-Mart Performance Summary

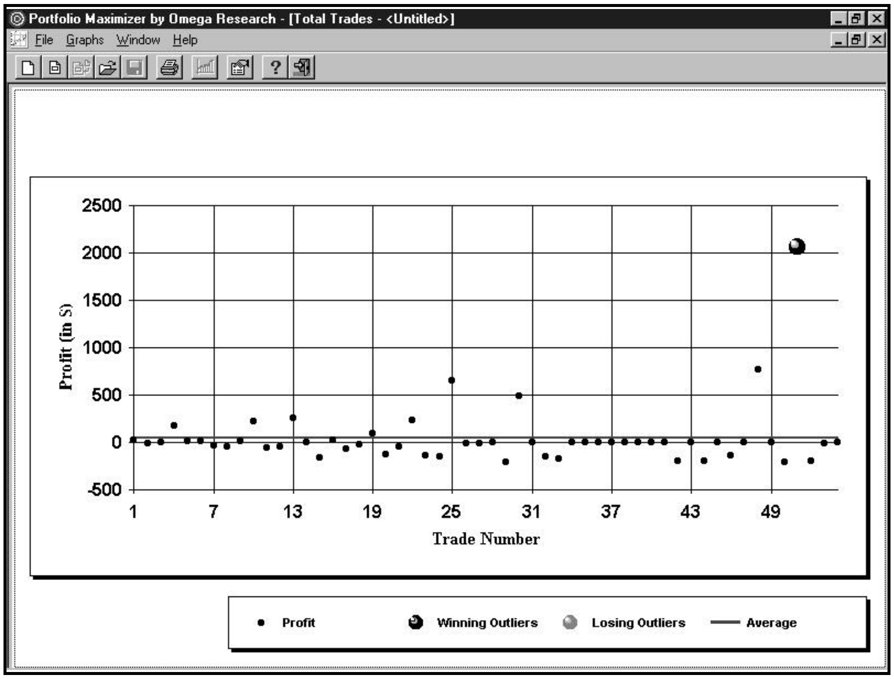
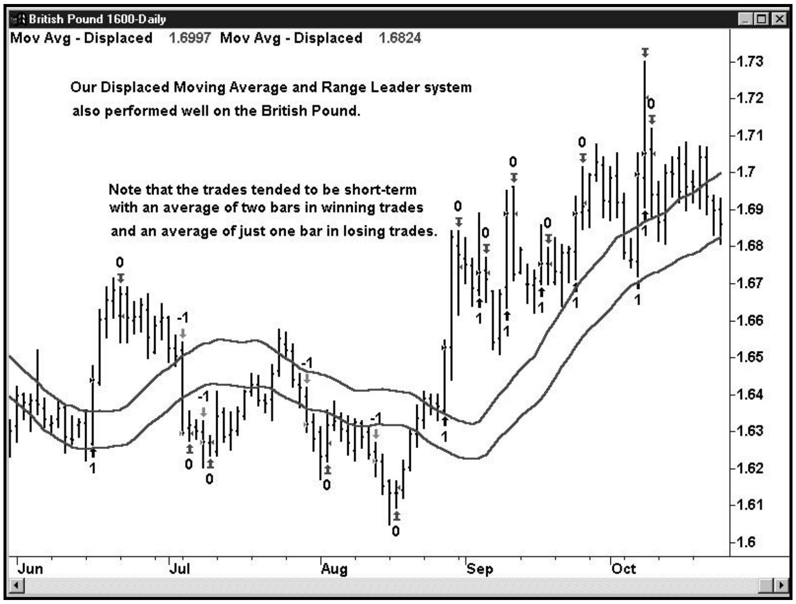


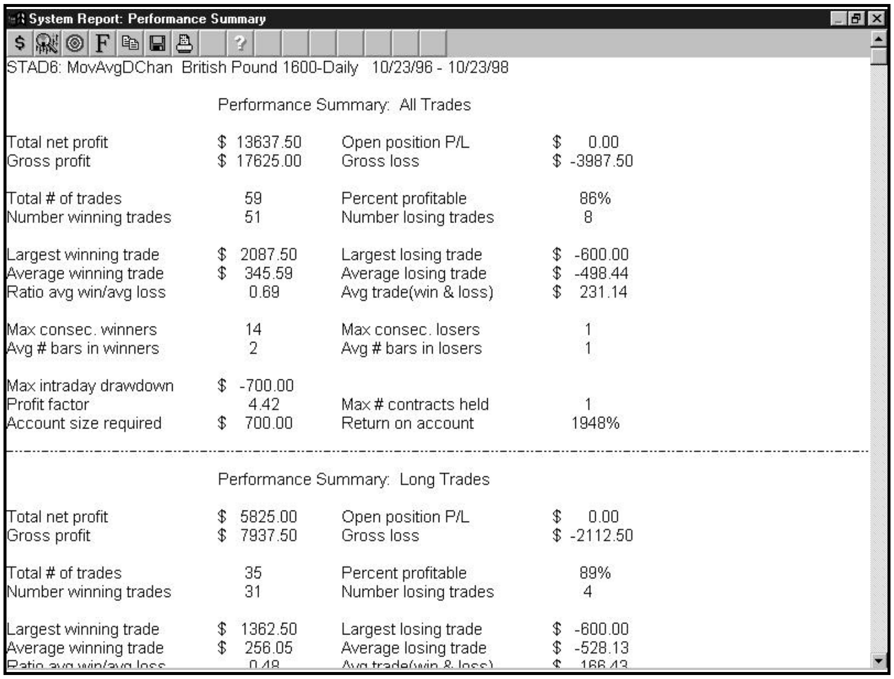
Figure 15 - Wal-Mart Total Trades Graph



Our Displaced Moving Average and Range Leader system also performed well on the British Pound.

Note that the trades tended to be short-term with an average of two bars in winning trades and an average of just one bar in losing trades.

Figure 16 - British Pound Daily Chart



```
STAD6: MovAvgDChan  British Pound 1600-Daily  10/23/96 - 10/23/98

                           Performance Summary:  All Trades

Total net profit            $ 13637.50    Open position P/L         $    0.00
Gross profit                $ 17625.00    Gross loss                $ -3987.50

Total # of trades               59        Percent profitable            86%
Number winning trades           51        Number losing trades           8

Largest winning trade       $  2087.50    Largest losing trade      $  -600.00
Average winning trade       $   345.59    Average losing trade      $  -498.44
Ratio avg win/avg loss          0.69      Avg trade(win & loss)     $   231.14

Max consec. winners             14        Max consec. losers             1
Avg # bars in winners            2        Avg # bars in losers           1

Max intraday drawdown       $  -700.00
Profit factor                   4.42      Max # contracts held           1
Account size required       $   700.00    Return on account           1948%
-------------------------------------------------------------------------------

                           Performance Summary:  Long Trades

Total net profit            $  5825.00    Open position P/L         $    0.00
Gross profit                $  7937.50    Gross loss                $ -2112.50

Total # of trades               35        Percent profitable            89%
Number winning trades           31        Number losing trades           4

Largest winning trade       $  1362.50    Largest losing trade      $  -600.00
Average winning trade       $   256.05    Average losing trade      $  -528.13
Ratio avg win/avg loss          0.48      Avg trade(win & loss)     $   166.43
```

Figure 17 - British Pound Performance Summary

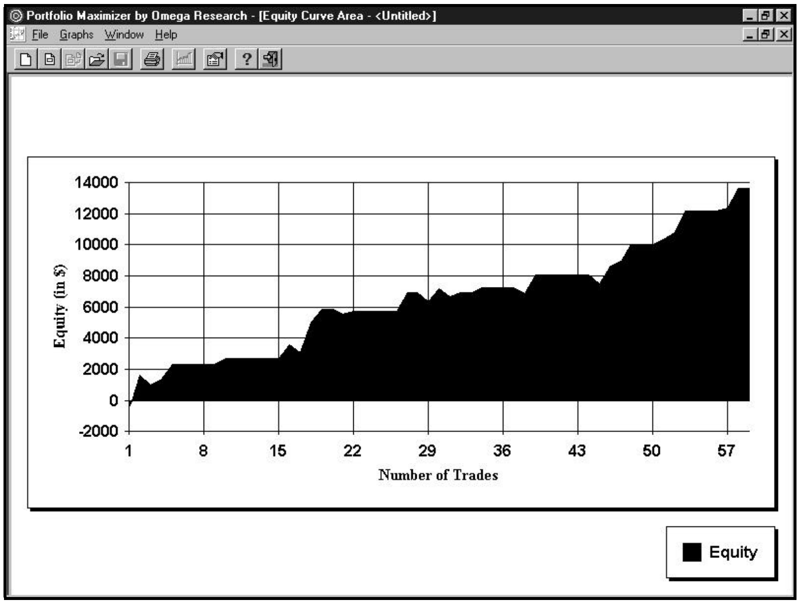


Figure 18 - British Pound Equity Curve Area Graph

Suggestions For Improvement

For a trend-following system, our Displaced Moving-Average Channel & Range Leaders System didn't generate many positive outliers. However, 57% winners on Wal-Mart and 86% winners on the British Pound are uncharacteristically high for a trend-following system. There is probably some room for improvement by sacrificing a little of the winning percent to allow profits to run and yield more positive outliers.

Four Sets of Moving Averages Crossover System

Almost every trader who practices a technical approach to trading uses moving averages in some way. Moving averages smooth price data to make the trend - and changes in the trend - easier to identify. Moving averages are also included as components of many popular technical indicators, such as ADX, MACD, Stochastic, Bollinger Bands, Keltner Channels, etc.

Many kinds of moving averages have been developed throughout the history of technical analysis. Some of the most widely used moving-average types are simple, weighted, exponential, displaced, and adaptive. In this system, we'll focus on simple moving averages.

A simple moving average (SMA) is calculated by adding the selected data field (most commonly the close) over a specified number of periods and dividing the total by the number of periods. A five-bar SMA of closes, for example, would add the five most recent closes and then divide the total by five.

Trading with moving averages can range from a simple to a complex methodology. A trader using one moving average could buy when a price bar closes above the moving average and sell when a bar closes below the moving average. A trader could use two moving averages (e.g. a five-bar average and a 20-bar average), buying when the faster average crosses above the slower one, and selling when the faster average crosses below the slower one. Another trader might prefer to use three moving averages - a fast (e.g. 4), a medium (e.g. 9), and a slow (e.g. 18). A buy setup would be in place when SMA:4 was above SMA:9, and SMA:9 was above SMA:18. A sell setup would be established with SMA:4 below SMA:9, and SMA:9 below SMA:18.

Our Four Sets of Moving Averages Crossover System is an attempt to improve on the dual moving average crossover system (DMAC) described above. DMAC assumes one moving-average combination is best for long entries, long exits, short entries, and short exits.

We wanted to find out if performance could be improved by optimizing the moving averages for each of the four tasks. Our tests showed that the values that worked best for one task (e.g. long entries) weren't the best-performing values for other tasks (e.g. long exits, short entries, and short exits). When a pair of moving averages was optimized for each task, net profit and other important performance fields improved considerably.

Defining Our Trading Rules

In this system, we defined separate long entries, short entries, long exits, and short exits. We also did some setup work to calculate the four sets of moving averages. The setup, entries, and exits are described next.

Setup

a) Using the MovAvg 2 lines indicator, calculate the following four pairs of simple moving averages:

Long entry = 7/23

Long exit = 4/8

Short entry = 6/18

Short exit = 2/9

Long Entries

a) The setup to buy is SMA:7 crosses above SMA:23.

b) The long entry is at the high of the setup bar plus one point.

Short Entries

a) The setup to sell short is SMA:6 crosses below SMA:18.

b) The short entry is at the low of the setup bar minus one point.

Exit Orders

a) Exit a long position on the next open when SMA:4 crosses below SMA:8.

b) Exit a short position on the next open when SMA:2 crosses above SMA:9.

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: Moving Average Pairs (STAD6: Mov Avg Pairs)

```
Inputs: LEFast(5), LESlow(20), LXFast(3), LXSlow(10), SEFast(5), SESlow(20), SXFast(3),
SXSlow(10);

{Long Entry/Exit}
IF Average(Close, LEFast) >  Average(Close, LESlow) Then
     Buy Next Bar at High + 1 Point Stop;
IF Average(Close, LXFast) Crosses Below Average(Close, LXSlow) Then
     ExitLong Next Bar at Market;

{Short Entry/Exit}
IF Average(Close, SEFast) < Average(Close, SESlow) Then
     Sell Next Bar at Low - 1 Point Stop;
IF Average(Close, SXFast) Crosses Above Average(Close, SXSlow) Then
     ExitShort Next Bar at Market;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
LEFast	5	Number of bars used to calculate the fast moving average for the Long Entry.
LESlow	20	Number of bars used to calculate the slow moving average for the Long Entry.
LXFast	3	Number of bars used to calculate the fast moving average for the Long Exit.
LXSlow	10	Number of bars used to calculate the slow moving average for the Long Exit.
SEFast	5	Number of bars used to calculate the fast moving average for the Short Entry.
SESlow	20	Number of bars used to calculate the slow moving average for the Short Entry.
SXFast	3	Number of bars used to calculate the fast moving average for the Short Exit.
SXSlow	10	Number of bars used to calculate the slow moving average for the Short Exit.

Since each average is specific to the entry to which it belongs, the declaration of variables is not necessary.

Long Entry/Exit

For the Long Entry, if the fast moving average crosses above the slow moving average, a buy order is placed at the High of the current bar plus 1 point. For the Long Exit, the fast moving average must cross below the slow moving average. When this occurs, a long exit order is placed at the open of the next bar (market).

```
IF Average(Close, LEFast) >  Average(Close, LESlow) Then
      Buy Next Bar at High + 1 Point Stop;
IF Average(Close, LXFast) Crosses Below Average(Close, LXSlow) Then
      ExitLong Next Bar at Market;
```

Short Entry/Exit

For the Short Entry, if the fast moving average crosses below the slow moving average, a sell order is placed at the Low of the current bar minus 1 point. For the Short Exit, the fast moving average must cross above the slow moving average. When this occurs, a short exit order is placed at the open of the next bar (market).

```
IF Average(Close, SEFast) < Average(Close, SESlow) Then
      Sell Next Bar at Low - 1 Point Stop;
IF Average(Close, SXFast) Crosses Above Average(Close, SXSlow) Then
      ExitShort Next Bar at Market;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a breakeven stop and a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option Do not allow multiple entries in the same direction. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We applied the Four Sets of Moving Averages Crossover System to daily Deutsche Mark data from 06/93 to 10/98. [Figure 19 - DM daily chart] The system generated 177 trades, of which 41% were profitable, for a net gain of $16,587. [Figure 20 - Performance Summary] The average winner ($1,060.78) divided by the average loser ($589.30) yielded a ratio of 1.81. The profit factor (1.27) indicates that the system made $1.27 for each dollar it lost. The system let profits run by staying in the average winner for ten bars, while it cut losses short by exiting from the average loser in only three bars.

We tested money-management stops from $800 to $1,200 in increments of $100. A $1,000 money-management stop produced the best results. [Figure 21 - Optimization graph]

In Portfolio Maximizer, we created a graph of the Total Trades. [Figure 22 - Total Trades graph] The striking feature of this graph is the four winning outliers versus no losing outliers (an outlier is a trade more than three standard deviations from an average trade).

We also tested this system (long side only) on weekly data of Merck from 01/78 to 10/98. The 50 trades generated a profit of $9,438. [Figure 23 - Performance Summary] Sixty percent of the trades were profitable, and the average trade earned $188. The average winning trade was 2.62 times as big as the average losing trade, and the system made $3.93 for each dollar it lost. Over the 20-year test period, there were eight consecutive winners, but only three consecutive losers. The system stayed with winning trades an average of 18 bars, while exiting losing trades in an average of ten bars.

The optimized parameters for weekly Merck data (buy only) were 5/20 for long entries and 3/10 for long exits. The equity curve shows a long, flat period through trade 24 but a rising period from trade 25 to trade 50. [Figure 24 - Equity Curve graph] Although we would all like to see our equity curves rise across the page at about a 45 degree angle, trend-following systems do not usually produce such equity curves. Our system neither made nor lost much money while Merck was fluctuating in a trading range, but it captured substantial profits when the stock began to trend higher.

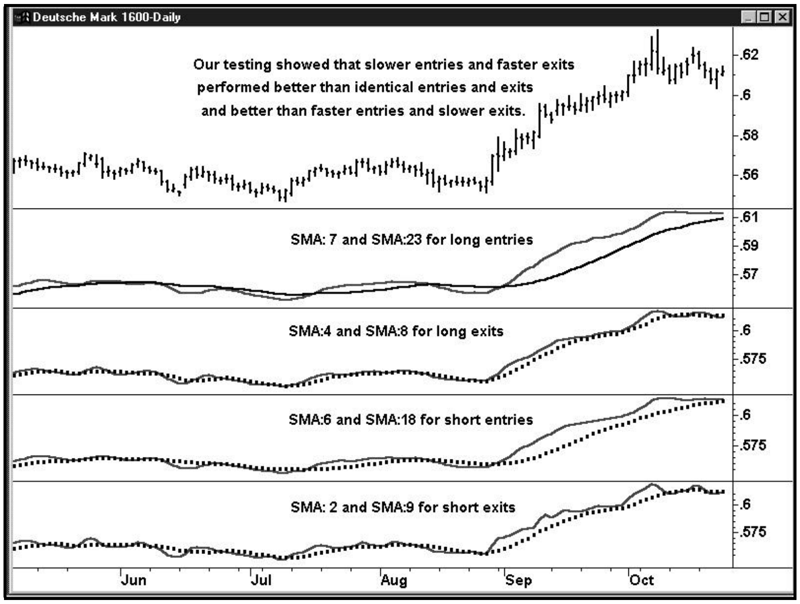
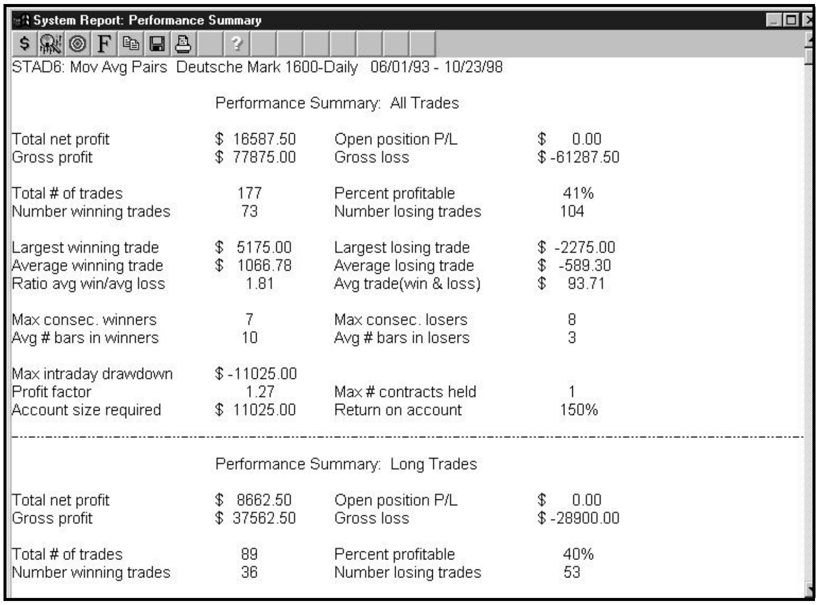

Our testing showed that slower entries and faster exits performed better than identical entries and exits and better than faster entries and slower exits.

Figure 19 - Deutsche Mark Daily Chart



```
STAD6: Mov Avg Pairs  Deutsche Mark 1600-Daily   06/01/93 - 10/23/98

                          Performance Summary:  All Trades

Total net profit          $  16587.50    Open position P/L          $      0.00
Gross profit              $  77875.00    Gross loss                 $ -61287.50

Total # of trades              177       Percent profitable              41%
Number winning trades           73       Number losing trades           104

Largest winning trade     $   5175.00    Largest losing trade       $  -2275.00
Average winning trade     $   1066.78    Average losing trade       $   -589.30
Ratio avg win/avg loss          1.81     Avg trade(win & loss)      $     93.71

Max consec. winners              7       Max consec. losers               8
Avg # bars in winners           10       Avg # bars in losers             3

Max intraday drawdown     $ -11025.00
Profit factor                   1.27     Max # contracts held             1
Account size required     $  11025.00    Return on account              150%
--------------------------------------------------------------------------------

                          Performance Summary:  Long Trades

Total net profit          $   8662.50    Open position P/L          $      0.00
Gross profit              $  37562.50    Gross loss                 $ -28900.00

Total # of trades               89       Percent profitable              40%
Number winning trades           36       Number losing trades            53
```

Figure 20 - Deutsche Mark Performance Summary

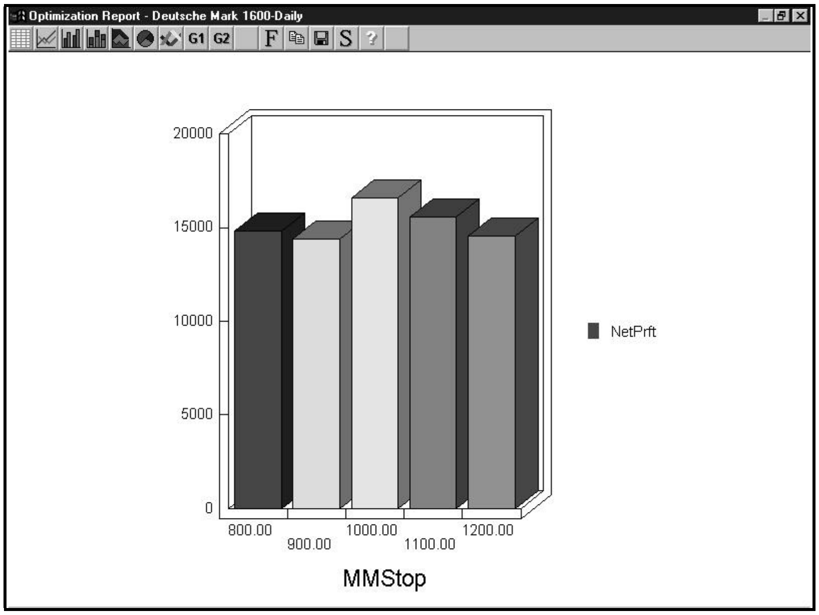


Figure 21 - Deutsche Mark Optimization Graph

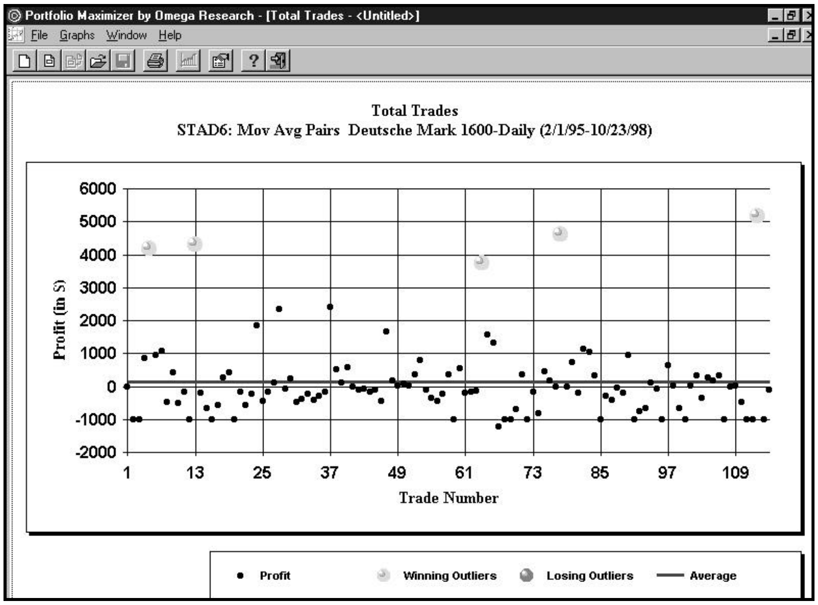
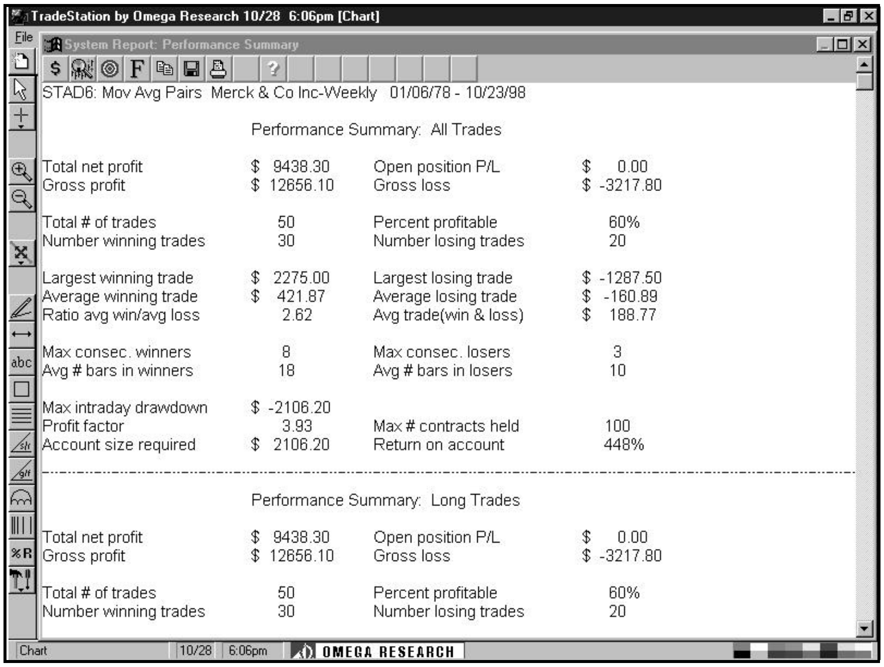


Figure 22 - Deutsche Mark Total Trades Graph


TradeStation by Omega Research 10/28 6:06pm [Chart]
File
System Report: Performance Summary
Chart
10/28
6:06pm
OMEGA RESEARCH

```
STAD6: Mov Avg Pairs  Merck & Co Inc-Weekly   01/06/78 - 10/23/98

                         Performance Summary:  All Trades

Total net profit          $   9438.30     Open position P/L        $      0.00
Gross profit              $  12656.10     Gross loss               $  -3217.80

Total # of trades              50         Percent profitable             60%
Number winning trades          30         Number losing trades           20

Largest winning trade     $   2275.00     Largest losing trade     $  -1287.50
Average winning trade     $    421.87     Average losing trade     $   -160.89
Ratio avg win/avg loss          2.62      Avg trade(win & loss)    $    188.77

Max consec. winners             8         Max consec. losers              3
Avg # bars in winners          18         Avg # bars in losers           10

Max intraday drawdown     $  -2106.20
Profit factor                   3.93      Max # contracts held          100
Account size required     $   2106.20     Return on account             448%
-----------------------------------------------------------------------------

                         Performance Summary:  Long Trades

Total net profit          $   9438.30     Open position P/L        $      0.00
Gross profit              $  12656.10     Gross loss               $  -3217.80

Total # of trades              50         Percent profitable             60%
Number winning trades          30         Number losing trades           20
```

Figure 23 - Merck & Co. Performance Summary

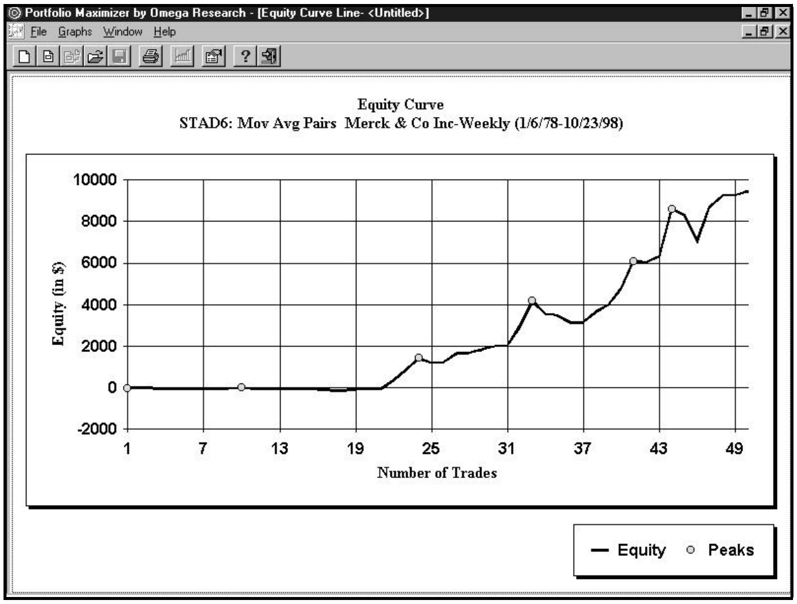


Figure 24 - Merck & Co. Equity Curve Graph

Suggestions For Improvement

The system as it stands treats all moving average crossovers equally. Although we wouldn't want to add any additional conditions to the exits, we think that the entries could be improved in two ways. The first condition we might add to our entries is that ADX (a "trendiness" indicator) must be rising on the bar that generates the moving-average crossover. This would insure that the market is in a trending rather than a sideways mode.

The second condition we might add to our entries uses volume to confirm the significance of the moving-average crossover. We could require that volume on the setup bar be greater than average volume. For example, if volume was greater than its ten-bar simple moving average, we would take the trade signalled by the moving-average crossover, but if volume was less than its ten-bar average, we would pass on the trade.

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 movements characterizing directionless markets.
- They have a high percentage of winning trades, with small profits from each winner.
- They are emotionally easier to trade than trend-following systems.

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 include trending systems and perhaps one or more volatility systems.

In this Chapter, we present two Support & Resistance systems, designed to make the most of the sideways movement of a market.

In This Chapter

Commodity Channel Index Average System

Our Commodity Channel Index Average System combines a support and resistance setup (buy low/sell high, sell high/buy low) with a price-channel-breakout entry, and a unique trailing stop.

The CCI indicator is designed to detect the beginning and the ending of price cycles by integrating a moving average with a divisor that reflects both actual and possible trading ranges. Next, CCI measures the market's deviation from normal in an attempt to discern a major change in trend. CCI Average applies another moving average to CCI to smooth its fluctuations. We draw the CCI Average as a histogram with an overbought line at +100 and an oversold line at -100.

Our setup to buy is the CCI Average crossing from below -100 to above -100. The long entry condition is an upside penetration of a price channel. The setup assures us that we are buying low, and the entry technique requires prices to be moving up when we initiate a long position. Similarly, our setup to sell short is the CCI Average crossing from above +100 to below +100. The short entry condition is a downside penetration of a price channel. The setup assures us that we are selling high, and the entry technique requires prices to be moving down when we initiate a short position. For both long and short trades, the setup is cancelled if the entry is not triggered within n bars of the setup. Also, a bullish setup is cancelled if CCI Average falls back below -100; a bearish setup is cancelled if CCI Average rises back above +100.

The system's trailing stop attempts to protect profits more aggressively than most other trailing stops do. In this support and resistance system, we're not trying to let profits run with loose trailing stops; instead, we're buying low / selling short high and locking in our gains as the market reaches overbought and oversold extremes.

For a long position, our trailing stop begins at an n-bar low minus one point. For each bar on which this market posts a new high (highest high since entry), the parameter for the n-bar low trailing stop decreases by one. For example, if the value of the n-bar low begins at n = 5, then decrease the n-bar low to n = 4 when the market makes a new high. Decrease it to n = 3 after the next new high, and so on. The only exception is that we won't decrease the number of bars considered in the trailing stop when a decrease would have resulted in our position being stopped out on a previous bar.

For a short position, our trailing stop begins at an n-bar high plus one point. For each bar on which the market posts a new low (lowest low since entry), the parameter for the n-bar high trailing stop decreases by one. If the value of the n-bar high begins at n = 6, for example, then decrease the n-bar high to n = 5 when the market makes a new low. Decrease it to n = 4 after the next new low, and so forth. However, don't decrease the number of bars considered in the trailing stop when a decrease would have resulted in our position being stopped out on a previous bar.

We'll also enable money-management and breakeven stops for the CCI Average System.

Defining Our Trading Rules

In this system, we defined both long and short entries as well as exit orders and enabled money-management and breakeven stops. We also did some setup work to calculate the CCI Average and the price channel. The setup, entries, and exits are described next.

Setup

a) Calculate a 30-bar CCI with a five-bar average.

b) Calculate a nine-bar price channel of highs and lows.

Long Entries

a) Identify a setup, with CCI Average crossing from below -100 to above -100.

b) Enter long at the highest high of the last nine bars plus one point.

c) If the entry is not triggered within four bars of the setup, or if CCI Average crosses back below -100, the setup is cancelled.

Short Entries

a) Identify a setup, with CCI Average crossing from above +100 to below +100.

b) Enter short at the lowest low of the last nine bars minus one point.

c) If the entry is not triggered within four bars of the setup, or if CCI Average crosses back above +100, the setup is cancelled.

Exit Orders

a) Our initial stop when long will be one point below the two-bar low; our initial stop when short will be one point above the two-bar high.

b) When long, we'll trail a stop at the most recent swing low (strength one); when short, we'll trail a stop at the most recent swing high (strength one).

c) We'll also enable a money-management stop, a breakeven stop, and a $ risk trailing stop.

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: Commodity Channel Index (STAD6: CCI System)

```
Input: CCILen(30), AvgLen(5), HiLoLen(9), EntryBar(4);
Vars: AvgCCI(0), BSetup(False), SSetup(False), TrailStop(0);

{Calculation of Average CCI and determination of setup conditions}
AvgCCI = Average(CCI(CCILen), AvgLen);
BSetup = AvgCCI Crosses Above -100;
SSetup = AvgCCI Crosses Below 100;

{Long Entry}
IF AvgCCI > -100 AND MRO(BSetup, EntryBar, 1) <> -1 Then
        Buy Next Bar at Highest(High, HiLoLen)[1] + 1 Point Stop;

{Short Entry}
IF AvgCCI < 100 AND MRO(SSetup, EntryBar, 1) <> -1 Then
        Sell Next Bar at Lowest(Low, HiLoLen)[1] - 1 Point Stop;
```

```
{Sets value for Lowest calculation used in Long Exit}
IF MarketPosition = 1 AND High > Highest(High, HiLoLen)[1] Then Begin
        IF TrailStop > 1 Then
                  TrailStop = TrailStop - 1;
End;

{Sets value for Highest calculation used in Short Exit}
IF MarketPosition = -1 AND Low < Lowest(Low, HiLoLen)[1] Then Begin
        IF TrailStop > 1 Then
                  TrailStop = TrailStop - 1;
End;

{Resets the calculation of the Exits for on the first bar of the position}
IF BarsSinceEntry = 0 Then
        TrailStop = HiLoLen;

{Exits}
ExitLong Next Bar at Lowest(Low, TrailStop) - 1 Point Stop;
ExitShort Next Bar at Highest(High, TrailStop) + 1 Point Stop;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
CCILen	30	Length, expressed in bars, used to calculate the CCI.
AvgLen	5	Length, expressed in bars, used to calculate the moving average of the CCI.
HiLoLen	9	Length, expressed in bars, used to calculate the Highest and Lowest prices used in for the determination of the entry price.
EntryBar	4	Number of bars for which the entry setup is valid.

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

```
Vars: AvgCCI(0), BSetup(False), SSetup(False), TrailStop(0);
```

Setup

In the setup we calculate the average CCI and store the resulting value in the variable AvgCCI. BSetup is a True/False variable that returns true when the AvgCCI value crosses above a value of -100. SSetup is a True/False variable that returns true when the AvgCCI value crosses below a value of 100.

```
AvgCCI = Average(CCI(CCILen), AvgLen);
BSetup = AvgCCI Crosses Above -100;
SSetup = AvgCCI Crosses Below 100;
```

Long Entry

A Long Entry is triggered when the AvgCCI value is greater than -100 and the cross above -100 occurred within a period of EntryBars ago. The MRO function is used to make the crossover determination. If the crossover occurred less than EntryBars ago, the function will return a value greater than or equal to 0, otherwise, if there was no crossover within EntryBars, the MRO function will return a -1. Thus we check that the MRO function is not equal to -1. If both conditions are found to be true, a long entry is generated at the highest High of the last HiLoLen bars, of 1 bar ago, plus 1 point.

```
IF AvgCCI > -100 AND MRO(BSetup, EntryBar, 1) <> -1 Then
        Buy Next Bar at Highest(High, HiLoLen)[1] + 1 Point Stop;
```

Short Entry

A Short Entry is triggered when the AvgCCI value is less than 100 and a crossover above 100 occurred within a period of EntryBars ago. The MRO function is used to make the crossover determination. If the crossover occurred less than EntryBars ago, the function will return a value greater than or equal to 0. If there was no crossover within EntryBars, the MRO function will return a -1. Thus, we check that the MRO function is not equal to -1. If both conditions are found to be true, a short entry is generated at the lowest Low of the last HiLoLen bars, of 1 bar ago, minus 1 point.

```
IF AvgCCI < 100 AND MRO(SSetup, EntryBar, 1) <> -1 Then
        Sell Next Bar at Lowest(Low, HiLoLen)[1] - 1 Point Stop;
```

Exit Setups

When the market position is Long, each new High causes the length of the lowest calculation, represented by the TrailStop variable, to be decreased by a value of 1. The TrailStop value may not decrease below 1. When the market position is Short, each new Low causes the length of the highest calculation, also represented by the variable TrailStop, to be decreased by a value of 1. The TrailStop value may not decrease below 1.

```
IF MarketPosition = 1 AND High > Highest(High, HiLoLen)[1] Then Begin
        IF TrailStop > 1 Then
                TrailStop = TrailStop - 1;
End;
IF MarketPosition = -1 AND Low < Lowest(Low, HiLoLen)[1] Then Begin
        IF TrailStop > 1 Then
                TrailStop = TrailStop - 1;
End;
```

On the first bar of a position, the TrailStop variable will always be reset to the original HiLoLen value.

```
IF BarsSinceEntry = 0 Then
        TrailStop = HiLoLen;
```

The System Exits are based on Highest/Lowest calculations with a TrailStop length. As the TrailStop value is adjusted in the prior section, the calculation of the Long and Short exit values are also adjusted.

```
ExitLong Next Bar at Lowest(Low, TrailStop) - 1 Point Stop;
ExitShort Next Bar at Highest(High, TrailStop) + 1 Point Stop;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a breakeven stop and a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option Do not allow multiple entries in the same direction. If the system is in a position and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We tested the CCI Average System on a weekly chart of United Technologies from 01/70 to 10/98. [Figure 1 - UTX weekly chart] The system made money on both the long and short sides, which has been unusual for our systems during the prolonged bull market in stocks. CCI Average earned $3,567 on 36 trades, 58% of which were profitable. The largest winning trade was $1,200, while the largest losing trade was only -$200. The system stayed with winners for an average of 13 bars but exited losers in an average of seven bars. The average winning trade was 2.71 times as big as the average losing trade, and the average trade (wins and losses) made $99.09 per hundred shares. The profit factor (dollars won per dollar lost) was excellent at 3.79. [Figure 2 - Performance Summary]

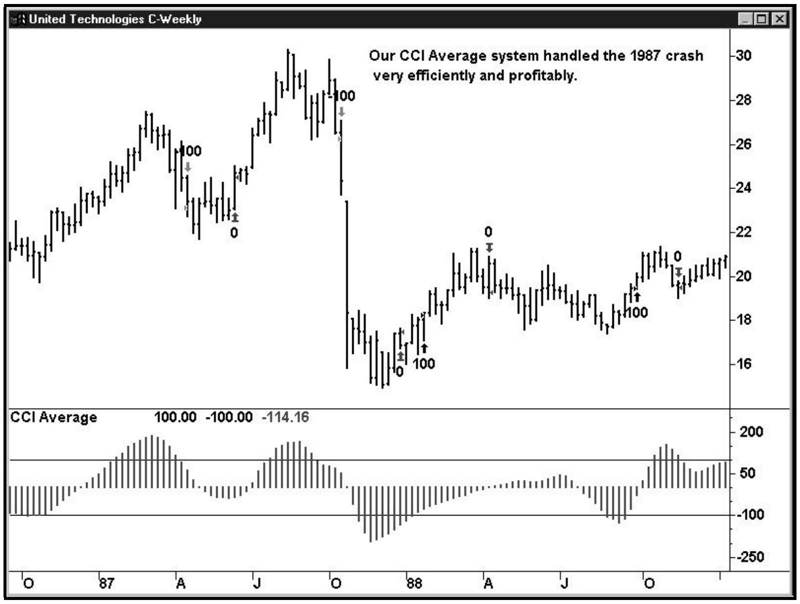
The Total Trades graph (created in Omega Research's Portfolio Maximizer) shows that there was one winning outlier (a trade with a gain of more than three standard deviations above the average winning trade) and no losing outliers. [Figure 3 - Total Trades graph] Since CCI Average is more of a support and resistance system than a trend-following system, we were pleased to see even one winning outlier.

We also tested the system on the CRB Index daily data from 01/95 to 10/98. [Figure 4 - CRB daily chart] We used the same parameters that we used on our test of United Technologies with the exception of decreasing the floor for the breakeven stop from $400 for weekly UTX to $100 for the daily CRB Index.

The total net profit (long and short) was $14,275 on 29 trades, with 86% profitable and an average trade of $492. [Figure 5 - Performance Summary] The profit factor of 5.76 means that our system made $5.76 for each dollar it lost.

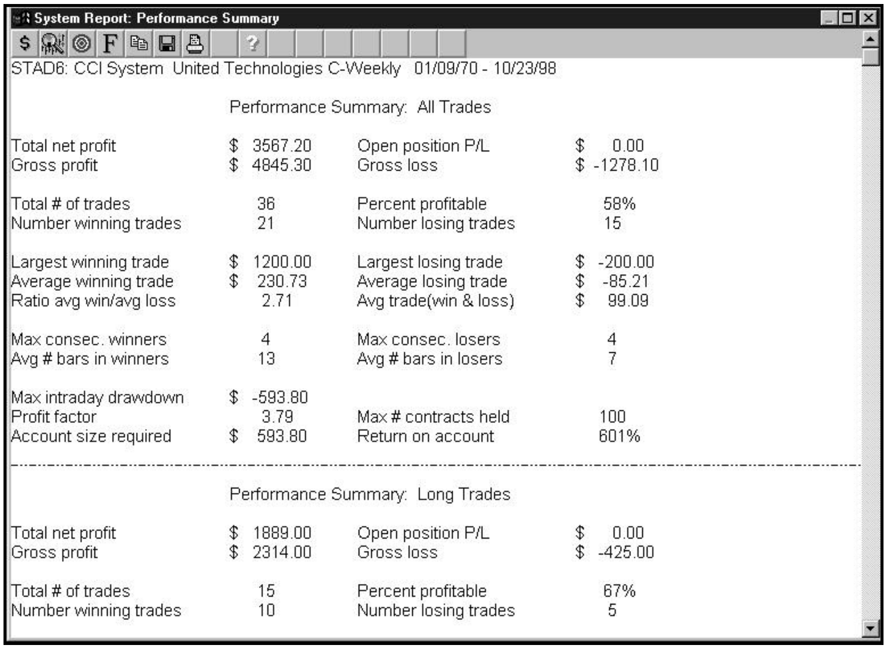
Over the roughly four-year period, the system generated 11 consecutive winning trades versus only two consecutive losing trades.

In Portfolio Maximizer, we created a graph of Monthly Rolling Net Profit. [Figure 6 - Monthly Rolling Net Profit graph] This graph is the result of a mark-to-market equity analysis at the end of each month. (Mark-to-market means that both open and closed profits and losses were included at month end.) The graph shows no major equity declines, but it does reveal two extended periods in which the system did not make new profits.



Our CCI Average system handled the 1987 crash very efficiently and profitably.

Figure 1 - UTX Weekly Chart



```
STAD6: CCI System  United Technologies C-Weekly   01/09/70 - 10/23/98

                         Performance Summary:  All Trades

Total net profit          $  3567.20     Open position P/L          $     0.00
Gross profit              $  4845.30     Gross loss                 $ -1278.10

Total # of trades              36        Percent profitable              58%
Number winning trades          21        Number losing trades            15

Largest winning trade     $  1200.00     Largest losing trade       $  -200.00
Average winning trade     $   230.73     Average losing trade       $   -85.21
Ratio avg win/avg loss          2.71     Avg trade(win & loss)      $    99.09

Max consec. winners             4        Max consec. losers               4
Avg # bars in winners          13        Avg # bars in losers             7

Max intraday drawdown     $  -593.80
Profit factor                   3.79     Max # contracts held           100
Account size required     $   593.80     Return on account              601%
-------------------------------------------------------------------------------
                         Performance Summary:  Long Trades

Total net profit          $  1889.00     Open position P/L          $     0.00
Gross profit              $  2314.00     Gross loss                 $  -425.00

Total # of trades              15        Percent profitable              67%
Number winning trades          10        Number losing trades             5
```

Figure 2 - UTX Performance Summary

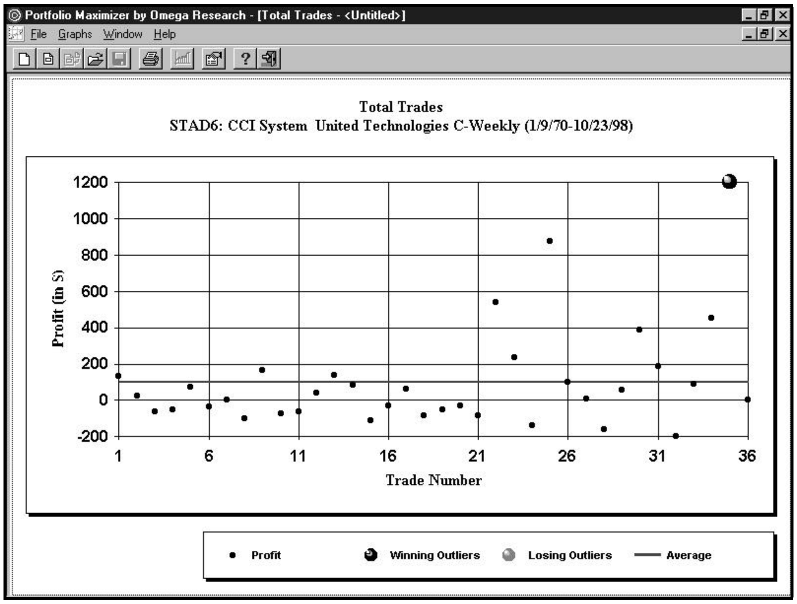
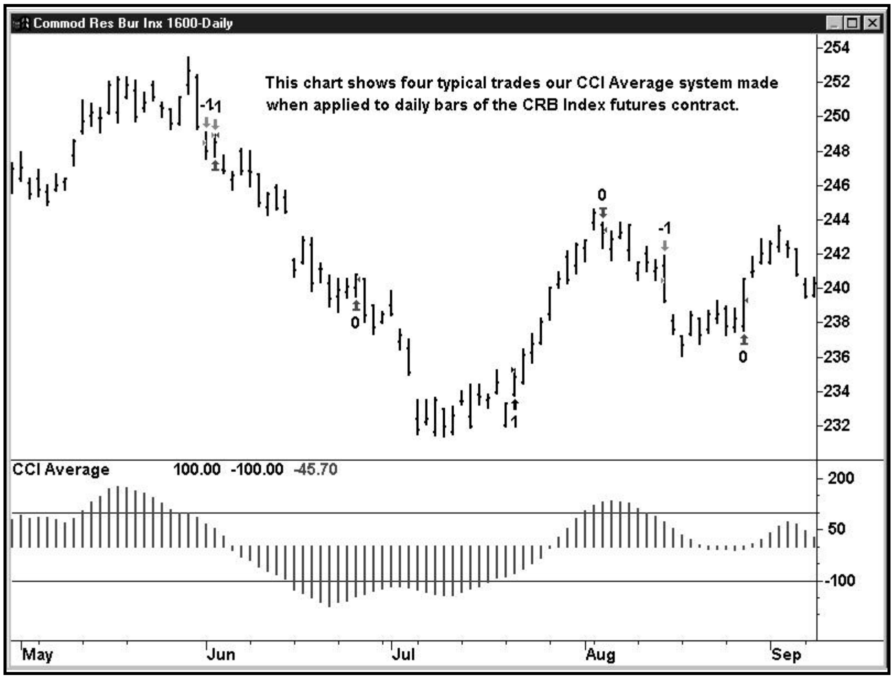
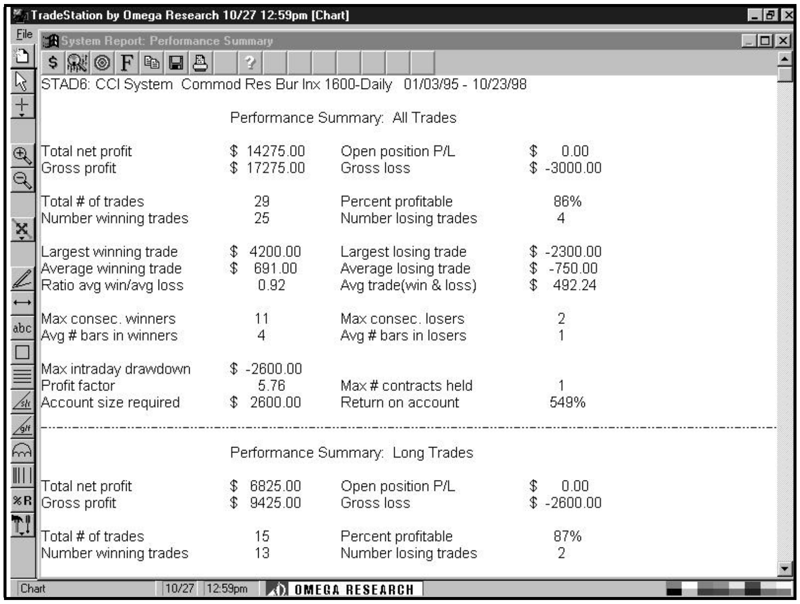


Figure 3 - UTX Total Trades Graph



This chart shows four typical trades our CCI Average system made when applied to daily bars of the CRB Index futures contract.

Figure 4 - CRB Index Daily Chart


TradeStation by Omega Research 10/27 12:59pm [Chart]
File
System Report: Performance Summary
Chart
10/27
12:59pm
OMEGA RESEARCH

```
STAD6: CCI System  Commod Res Bur Inx 1600-Daily   01/03/95 - 10/23/98

                          Performance Summary:  All Trades

Total net profit          $  14275.00     Open position P/L         $      0.00
Gross profit              $  17275.00     Gross loss                $  -3000.00

Total # of trades               29        Percent profitable              86%
Number winning trades           25        Number losing trades             4

Largest winning trade     $   4200.00     Largest losing trade      $  -2300.00
Average winning trade     $    691.00     Average losing trade      $   -750.00
Ratio avg win/avg loss          0.92      Avg trade(win & loss)     $    492.24

Max consec. winners             11        Max consec. losers               2
Avg # bars in winners            4        Avg # bars in losers             1

Max intraday drawdown     $  -2600.00
Profit factor                   5.76      Max # contracts held             1
Account size required     $   2600.00     Return on account              549%
-------------------------------------------------------------------------------
                          Performance Summary:  Long Trades

Total net profit          $   6825.00     Open position P/L         $      0.00
Gross profit              $   9425.00     Gross loss                $  -2600.00

Total # of trades               15        Percent profitable              87%
Number winning trades           13        Number losing trades             2
```

Figure 5 - CRB Index Performance Summary

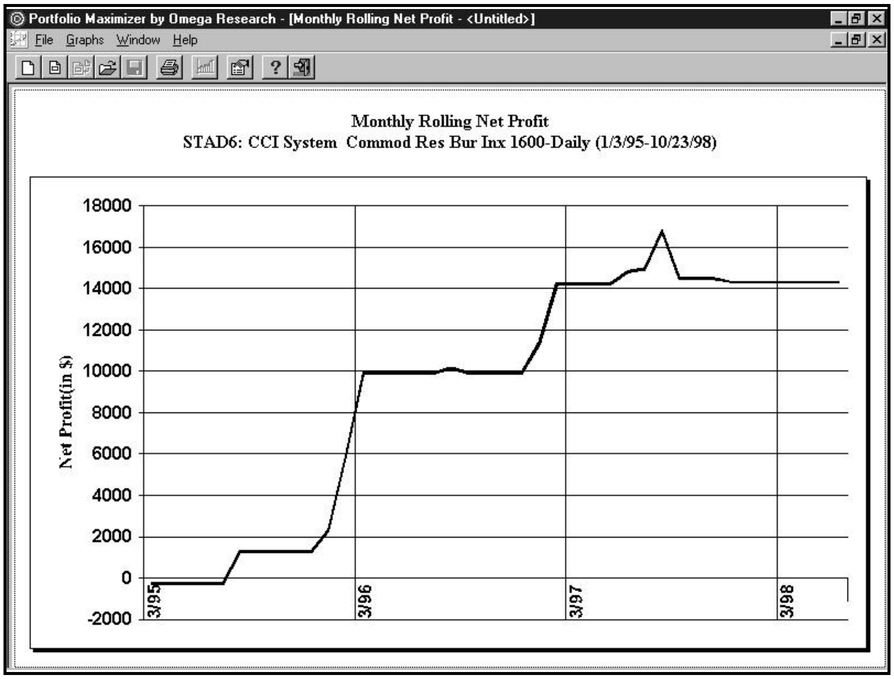


Figure 6 - CRB Index Monthly Net Profit Graph

Suggestions For Improvement

We accepted the default values of +100 for overbought and -100 for oversold without testing to see if they are the best possible parameters. It's possible that optimized values would have yielded even better test results. Of course, the optimal values would not have to be identical except for the + or - sign. For example, optimization might reveal that +90 is the strongest value for overbought and -70 is the best-performing value for oversold. By the way, the previous suggestion is the kind of optimization that we believe in and advocate at Omega Research. Optimization should be one of the last steps in developing a trading system. The proper use of optimization is to fine-tune a system that is already producing good test results, not to turn a losing system into a winner.

Minor Reaction/Trend Resumption

The Minor Reaction/Trend Resumption System (MR/TR) was designed by Jack Schwager and published in his book Schwager on Futures: Technical Analysis (John Wiley & Sons, 1996). Schwager called this system "Reversal of Minor Reaction", and included it as an example (for illustrative purposes only) of midtrend entry and pyramiding.

MR/TR uses a price channel to determine the trend. If the most recent breakout of the channel was to the upside, the trend is bullish; if the most recent breakout was to the downside, the trend is bearish.

If the market is in an uptrend, we begin monitoring for a minor decline to support; in a downtrend, we monitor for a minor rally to resistance. The reaction count begins at zero. In an uptrend, it's raised to one when the current bar's high and low are less than the corresponding high and low on the bar that made the high of the uptrend. In a downtrend, the reaction count is raised to one when the current bar's high and low are greater than the corresponding high and low on the bar that made the low of the downtrend.

The reaction count in an uptrend increases by one for each bar on which the high and low are less than the corresponding points of the most recent bar on which the reaction count was increased. In a downtrend, the reaction count increases by one for each bar on which the high and low are greater than the corresponding points of the most recent bar on which the reaction count was increased. The reaction count is set back to zero any time the market makes a new high for the uptrend or a new low for the downtrend.

When the reaction count reaches four, we begin monitoring for thrust bars. In an uptrend, a thrust bar closes above the previous bar's high; in a downtrend, a thrust bar closes below the previous bar's low. The thrust count is set back to zero any time the market falls below the previous low of the reaction in an uptrend; the thrust count is set back to zero in a downtrend any time the market rallies above the previous high of the reaction.

When the thrust count reaches two, a setup is complete. In an uptrend, we'll buy at the high of the setup bar plus one point; in a downtrend, we'll sell short at the low of the setup bar minus one point.

The next step is to place both the initial stop and the trailing stop. We'll place our initial stop one point below the low of the reaction in an uptrend, and we'll trail a $ risk trailing stop. In a downtrend, we'll place our initial stop one point above the high of the reaction and trail a $ risk trailing stop.

Defining Our Trading Rules

In this system, we defined long entries, short entries, and initial protective stops. We also enabled a $ risk trailing stop. We did some setup work to calculate the price channel. The setup, entries, and exits are described next.

Setup

a) Calculate a 40-bar price channel of highs and lows.

Long Entries

a) Determine the trend by locating the most recent breakout above or below the price channel. If the breakout was above the channel, begin monitoring for downward reaction bars. When the reaction count reaches four, begin monitoring for bullish thrust bars. When the thrust count reaches two, the buy setup is complete.

b) Buy at the high of the setup bar plus one point. The setup is cancelled if the entry is not triggered within four bars.

Short Entries

a) Determine the trend by locating the most recent breakout above or below the price channel. If the breakout was below the channel, begin monitoring for upward reaction bars. When the reaction count reaches four, begin monitoring for bearish thrust bars. When the thrust count reaches two, the sell setup is complete.

b) Sell short at the low of the setup bar minus one point. The setup is cancelled if the entry is not triggered within four bars.

Exit Orders

a) Place a protective stop one point below the low of the reaction for a long position and one point above the high of the reaction for a short position.

b) We'll also enable a $ risk trailing stop.

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: Minor Reversal/Trend Resumption (STAD6: Trend Rev/Res)

```
Inputs: ChanLen(40), Reaction(4), TCount(2), Setup(4);
Vars: HiChan(0), LoChan(0), LReaction(0), LReacHigh(0), LReacLow(0), SReaction(0),
SReacHigh(0), SReacLow(0), LThrust(0), SThrust(0), LEntry(0), LExit(0), SEntry(0), SExit(0),
NewLThrust(False), NewSThrust(False), LTR(False), STR(False);

{Assignment of values for the High/Low Channel}
HiChan = Highest(High, ChanLen)[1];
LoChan = Lowest(Low, ChanLen)[1];

{Reversal Breakout - Initial Setup}
IF MarketPosition <> 1 Then Begin
    IF High > HiChan Then Begin
         LReaction = 1;
         LReacHigh = High;
         LReacLow = Low;
         LExit = LReacLow;
         LThrust = 0;
         SReaction = 0;
    End;
End
Else Begin
    LReaction = 0;
    LThrust = 0;
End;

{Reversal Breakout - Initial Setup}
IF MarketPosition <> -1 Then Begin
    IF Low < LoChan Then Begin
```

```
                SReaction = 1;
                SReacHigh = High;
                SReacLow = Low;
                SExit = SReacLow;
                SThrust = 0;
                LReaction = 0;
        End;
End
Else Begin
        SReaction = 0;
        LThrust = 0;
End;

{Upward Trend Resumption}
IF LReaction >= 1 Then Begin
        IF Low < LExit Then Begin
                LExit = Low;
                LThrust = 0;
        End;
        IF High < LReacHigh AND Low < LReacLow Then Begin
                LReaction = LReaction + 1;
                LReacHigh = High;
                LReacLow = Low;
                NewLThrust = True;
        End;
End;

{Downward Trend Resumption}
IF SReaction >= 1 Then Begin
        IF High > SExit Then Begin
                SExit = High;
                SThrust = 0;
        End;
        IF Low > SReacLow AND High > SReacHigh Then Begin
                SReaction = SReaction + 1;
                SReacLow = Low;
                SReacHigh = High;
                NewSThrust = True;
        End;
End;

{Bullish Thrust Day accumulation}
IF LReaction >= Reaction AND Close > High[1] Then
        LThrust = LThrust + 1;

{Bearish Thrust Day accumulation}
IF SReaction >= Reaction AND Close < Low[1] Then
        SThrust = SThrust + 1;

{Thrust Count Completion}
LTR = LThrust = TCount AND NewLThrust;
STR = SThrust = TCount AND NewSThrust;

{Entry Price Setup}
IF LTR Then Begin
        LEntry = High + 1 Point;
        NewLThrust = False;
```

```
End;
IF STR Then Begin
        SEntry = Low - 1 Point;
        NewSThrust = False;
End;

{Entries}
IF MRO(LTR, Setup, 1) <> -1 AND LReaction >= Reaction Then
        Buy Next Bar at LEntry Stop;
IF MRO(STR, Setup, 1) <> -1 AND SReaction >= Reaction Then
        Sell Next Bar at SEntry Stop;

{Exits}
IF LReaction = 0 Then
        ExitLong Next Bar at LExit Stop;
IF SReaction = 0 Then
        ExitShort Next Bar at SExit Stop;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
ChanLen	40	Length, expressed in bars, used to calculate the High & Low Channel for the trend.
Reaction	4	Number of Reaction bars required for an Entry Setup
Tcount	2	Number of Thrust bars required for an Entry Setup
Setup	4	Number of bars for which an Entry is valid.

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

```
Vars: HiChan(0), LoChan(0), LReaction(0), LReacHigh(0), LReacLow(0), SReaction(0),
SReacHigh(0), SReacLow(0), LThrust(0), SThrust(0), LEntry(0), LExit(0), SEntry(0), SExit(0),
NewLThrust(False), NewSThrust(False), LTR(False), STR(False);
```

Setup

The Highest High and Lowest Low calculation is used to identify the direction of the trend. The calculation for each is assigned to the variables HiChan and LoChan respectively. Notice that both channel calculations are offset by 1 bar. Since we are looking for a breakout, we must compare the High/Low to the prior highest and lowest values, since there can never be a value higher than the current highest for any given bar, and vice versa.

```
HiChan = Highest(High, ChanLen)[1];
LoChan = Lowest(Low, ChanLen)[1];
```

If the current position is not long (0 or -1), we look for an uptrend breakout to begin the long reaction count. If a High greater than the HiChan value is identified, the LReaction variable is set to 1. In addition, the long Reaction bar High and Low are assigned to the variables LReacHigh and LReacLow. This is necessary for the identification of forthcoming Reaction bars. The LExit variable is set equal to the Low of the reaction bar to serve as the initial protective stop in the event that there are no lower Lows on subsequent long reaction bars. Finally, the variable for the long thrust count (LThrust) and the variable for the short reaction count (SReaction) are reset to 0. If the market position is equal to 1 (Long), then the long reaction count (LReaction) and the long thrust count are reset to 0. This assures that only one long entry can occur as a result of a set of breakout, reaction, and thrust bars.

```
IF MarketPosition <> 1 Then Begin
        IF High > HiChan Then Begin
                LReaction = 1;
                LReacHigh = High;
                LReacLow = Low;
                LExit = LReacLow;
                LThrust = 0;
                SReaction = 0;
        End;
End
Else Begin
        LReaction = 0;
        LThrust = 0;
End;
```

If the current position is not short (0 or 1), we look for a down trend breakout to begin the short reaction count. If a Low less than the LoChan value is identified, the SReaction variable is set to 1. In addition, the low Reaction bar High and Low are assigned to the variables LReacHigh and LReacLow. This is necessary for the identification of forthcoming Reaction bars. The SExit variable is set equal to the High of the reaction bar to serve as the initial protective stop in the event that there are no higher Highs on subsequent short reaction bars. Finally, the variable for the long thrust count (LThrust) and the variable for the short reaction count (SReaction) are reset to 0. If the market position is equal to -1 (Short), then the short reaction count (SReaction) and the short thrust count are reset to 0. This assures that only one short entry can occur as a result of a set of breakout, reaction, and thrust bars.

```
IF MarketPosition <> -1 Then Begin
        IF Low < LoChan Then Begin
                SReaction = 1;
                SReacHigh = High;
                SReacLow = Low;
                SExit = SReacLow;
                SThrust = 0;
                LReaction = 0;
        End;
End
Else Begin
        SReaction = 0;
        LThrust = 0;
End;
```

If the LReaction variable is greater than or equal to 1, meaning that we are currently looking for additional long reaction bars, we begin the process of redefining the exit and/or redefining the most recent long reaction bar as necessary. If the current Low is less than the LExit value that was set in the prior section, the LExit value is redefined as the current Low. The long thrust count is also assigned a value of 0. If a new long reaction bar occurs, in other words, the High is less than the LReacHigh value and the Low is also less than the LReacLow value, then we increment the long reaction counter. LReaction is incremented by a value of 1. In addition, the High and Low of the most recent reaction bar are assigned to the variables LReachHigh and LReacLow. The NewLThrust variable is set to true so as to indicate a green light for accumulation of thrust bars once the long reaction criterion has been met.

```
IF LReaction >= 1 Then Begin
    IF Low < LExit Then Begin
        LExit = Low;
        LThrust = 0;
    End;
    IF High < LReacHigh AND Low < LReacLow Then Begin
        LReaction = LReaction + 1;
        LReacHigh = High;
        LReacLow = Low;
        NewLThrust = True;
    End;
End;
```

If the SReaction variable is greater than or equal to 1, meaning that we are currently looking for additional short reaction bars, we begin the process of redefining the exit and/or redefining the most recent short reaction bar as necessary. If the current High is greater than the SExit value that was set in the prior section, the SExit value is redefined as the current High. The short thrust count is also assigned a value of 0. If a new short reaction bar occurs, in other words, the High is greater than the SReacHigh value and the Low is also greater than the SReacLow value, then we increment the short reaction counter. SReaction is incremented by a value of 1. In addition, the High and Low of the most recent reaction bar are assigned to the variables SReachHigh and SReacLow. The NewSThrust variable is set to true so as to indicate a green light for accumulation of thrust bars once the short reaction criterion has been met.

```
IF SReaction >= 1 Then Begin
    IF High > SExit Then Begin
        SExit = High;
        SThrust = 0;
    End;
    IF Low > SReacLow AND High > SReacHigh Then Begin
        SReaction = SReaction + 1;
        SReacLow = Low;
        SReacHigh = High;
        NewSThrust = True;
    End;
End;
```

If the long reaction counter (LReaction) is equal to or greater than the required Reaction value (as specified by the Inputs), we can begin to count the thrust bars. If the Close is greater than the High of the previous bar, the LThrust variable value is incremented by 1.

```
IF LReaction >= Reaction AND Close > High[1] Then
    LThrust = LThrust + 1;
```

If the short reaction counter (SReaction) is equal to or greater than the required Reaction value (as specified by the Inputs), we can begin to count the thrust bars. If the Close is less than the Low of the previous bar, the SThrust variable value is incremented by 1.

```
IF SReaction >= Reaction AND Close < Low[1] Then
        SThrust = SThrust + 1;
```

When the long thrust counter (LThrust) equals the thrust count value TCount (as specified by the Inputs), and the NewLThrust variable is True, the long trend resumption variable (LTR) is set to True. When the short thrust counter (SThrust) equals the thrust count value TCount (as specified by the Inputs), and the NewSThrust variable is True, the short trend resumption variable (STR) is set to True.

```
LTR = LThrust = TCount AND NewLThrust;
STR = SThrust = TCount AND NewSThrust;
```

If the long trend resumption variable (LTR) is True, the long entry price is set and NewLThrust is made False. The entry price is based on the current High, plus 1 point. Since the required number of thrust bars was met, the NewLThrust variable will have no use until the next set of long reaction bars is identified.

```
IF LTR Then Begin
        LEntry = High + 1 Point;
        NewLThrust = False;
End;
```

If the short trend resumption variable (STR) is True, the short entry price is set and NewSThrust is made False. The entry price is based on the current Low, minus 1 point. Since the required number of thrust bars was met, the NewSThrust variable will have no use until the next set of short reaction bars is identified.

```
IF STR Then Begin
        SEntry = Low - 1 Point;
        NewSThrust = False;
End;
```

Long Entry

For the long entry, the MRO function is used to determine if the LTR was True within the last Setup number of bars. Also, notice that the LReaction variable is once again being compared to the Reaction Input value. This is done in order to prevent multiple long entries as a result of the same buy setup. As was shown in a prior section, if the market position is long, the LReaction variable will be reset, thus preventing any additional long entries until a complete buy setup occurs.

```
IF MRO(LTR, Setup, 1) <> -1 AND LReaction >= Reaction Then
        Buy Next Bar at LEntry Stop;
```

Short Entry

For the short entry, the MRO function is used to determine if the STR was True within the last Setup number of bars. Also, notice that the SReaction variable is once again being compared to the Reaction Input value. This is done in order to prevent multiple short entries as a result of the same sell setup. As was shown in a prior section, if the market position is short, the SReaction variable will be reset, thus preventing any additional long entries until a complete sell setup occurs.

```
IF MRO(STR, Setup, 1) <> -1 AND SReaction >= Reaction Then
        Sell Next Bar at SEntry Stop;
```

Exits

As stated above, once a position has been taken, the reaction counters (LReaction and Sreaction) will be reset to 0. Once they have been reset, they then trigger the Exits at the LExit and SExit values that were calculated during the reaction counting stage.

```
IF LReaction = 0 Then
        ExitLong Next Bar at LExit Stop;
IF SReaction = 0 Then
        ExitShort Next Bar at SExit Stop;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We applied MR/TR to Hewlett Packard weekly data from 01/78 to 10/98 for both the long and short sides of the market. [Figure 7 - HP weekly chart] The system made $3,175 per 100 shares on 11 trades, with 64% profitable. [Figure 8 - Performance Summary] The largest winning trade was $1,725 compared to a largest losing trade of -$187. With an average winner of $534 and an average loser of $141, the ratio of average win to average loss was 3.8. The average trade earned $288. In the 20-year test period, the system generated six consecutive winners versus three consecutive losers. At 6.64 (dollars won per dollar lost), the profit factor was excellent. [Figure 9 - Equity Curve graph]

We also tested MR/TR on daily data of the Goldman Sachs Index from 06/93 to 10/98. The system earned $5,500 per contract on 23 trades, with 48% winners. The largest winning trade ($3,225) was more than ten times the size of the largest losing trade (-$300). With a $700 average winning trade and a $183 average losing trade, the average win/average loss was very good at 3.82. The profit factor was also very good at 3.50 (dollars won per dollar lost).

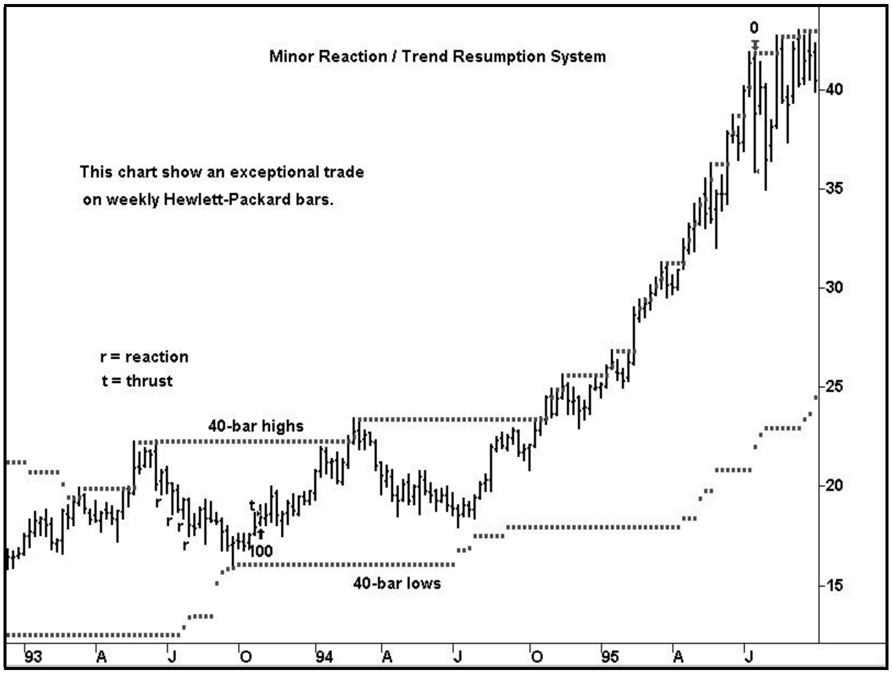
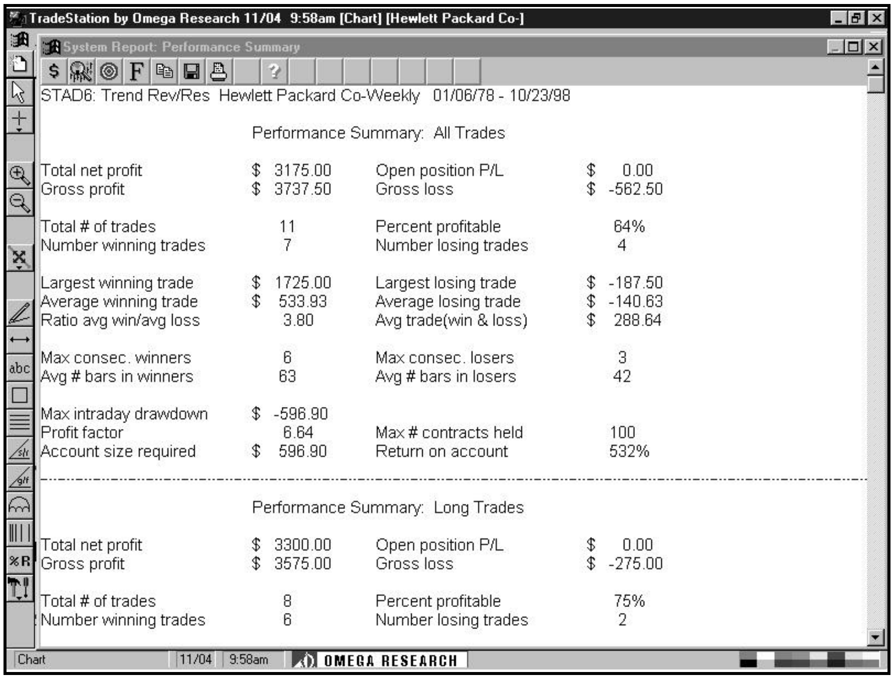


Figure 7- Hewlett Packard Weekly Chart


TradeStation by Omega Research 11/04 9:58am [Chart] [Hewlett Packard Co-]
System Report: Performance Summary
$
F
?
abc
%R
Chart
11/04
9:58am
OMEGA RESEARCH

```
STAD6: Trend Rev/Res  Hewlett Packard Co-Weekly   01/06/78 - 10/23/98

                         Performance Summary:  All Trades

Total net profit            $  3175.00     Open position P/L          $    0.00
Gross profit                $  3737.50     Gross loss                 $  -562.50

Total # of trades               11         Percent profitable            64%
Number winning trades            7         Number losing trades           4

Largest winning trade       $  1725.00     Largest losing trade       $  -187.50
Average winning trade       $   533.93     Average losing trade       $  -140.63
Ratio avg win/avg loss          3.80       Avg trade(win & loss)      $   288.64

Max consec. winners              6         Max consec. losers             3
Avg # bars in winners           63         Avg # bars in losers          42

Max intraday drawdown       $  -596.90
Profit factor                   6.64       Max # contracts held         100
Account size required       $   596.90     Return on account            532%
--------------------------------------------------------------------------------

                         Performance Summary:  Long Trades

Total net profit            $  3300.00     Open position P/L          $    0.00
Gross profit                $  3575.00     Gross loss                 $  -275.00

Total # of trades                8         Percent profitable            75%
Number winning trades            6         Number losing trades           2
```

Figure 8 - Hewlett Packard Performance Summary

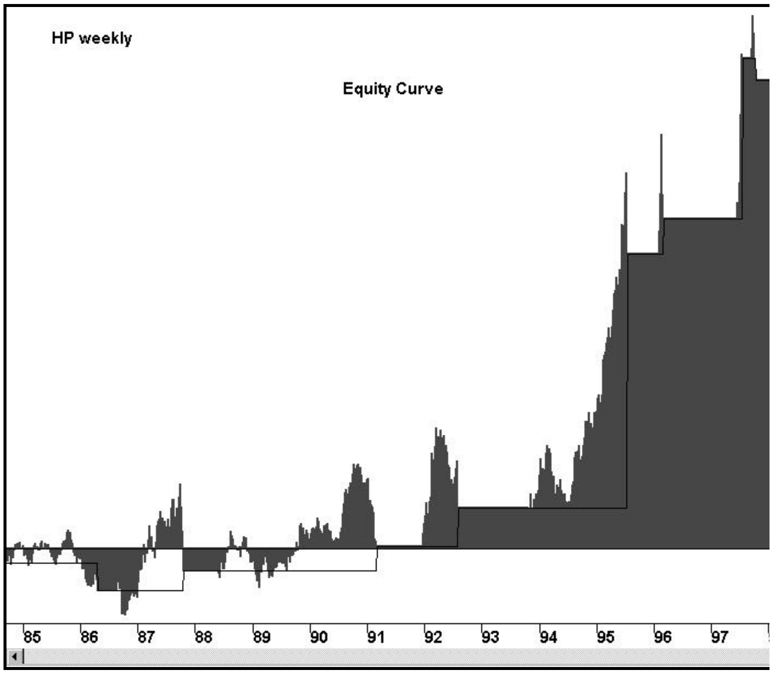


Figure 9 - Hewlett Packard Equity Curve Graph

Suggestions For Improvement

We would like to suggest some refinements to the definitions of reaction bars and thrust bars. For a reaction bar in an uptrend, try adding the requirement that the close must be less than the close of the most recent bar on which the reaction count was increased. In a downtrend, try adding the requirement that the close of a reaction bar must be greater than the close of the most recent bar on which the reaction count was increased.

Also, for thrust bars in both uptrends and downtrends, try requiring that the true range of the thrust bar be larger than the true range of the previous bar, and that volume for the thrust bar be above a ten-bar exponential moving average of volume.

These stricter requirements for reaction bars and thrust bars might increase their reliability and help to increase profits.

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 price action. Volatility breakout systems generally have the following characteristics:

- They spend a substantial amount of time out of the market.
- They generate a high percentage of winning trades but earn a small average profit per trade.
- They don't take advantage of big moves.
- They are exciting to trade because trades are quick and short-term.
- They are based primarily on increases in volatility.

When designing this type of system, the key is to anticipate and take advantage of a significant change in volatility and to exit the position without giving back a lot of open profits. In this chapter, we present three new volatility breakout systems.

In This Chapter

First Pullback System

Our First Pullback System (FPS) employs a sensitive MACD indicator to define the trend and to identify the first pullback in each trend. Linda Bradford Raschke, an exceptional trader and one of Jack Schwager's New Market Wizards popularized a variation of the MACD pattern we'll present in this section.

First, let's discuss the MACD indicator. MACD stands for Moving-Average Convergence/Divergence. It was created by Gerald Appel, a noted author and trader. The indicator consists of two lines - the MACD line and the Signal line. The MACD line is the difference between two exponential moving averages. In TradeStation, the default values are 12 and 26, so the MACD line is the spread between EMA:12 and EMA:26. The MACD line shows us which EMA is above the other, how far apart they are, and whether they're pulling farther apart or drawing closer together. The Signal line is a slowed down, less sensitive derivative of the MACD line. It's calculated by applying an EMA to the MACD line. The default value is nine, which means that the Signal line is a nine-bar EMA of the MACD line.

In FPS, we set the MACD values to a faster 4, 10, 16. We use the Signal line to define the trend, and the MACD line to identify the first pullback in each trend. To determine if we should be trying to buy a market or to sell it, we look first at the Signal line. If its most recent crossing of the zero line was from below to above, the trend is up; if the Signal line's most recent crossing of the zero line was from above to below, the trend is down.

In a new uptrend, we wait for the MACD line to cross below the Signal line for the first time since the Signal line crossed above zero. That completes the setup. The buy entry is at the close of the previous bar plus a percentage of an Average True Range. The setup remains in effect until either the entry is triggered or the Signal line crosses below zero.

In a new downtrend, we wait for the MACD line to cross above the Signal line for the first time since the Signal line crossed below zero. That completes the setup. The entry price for selling short is the close of the previous bar minus a percentage of an Average True Range. The setup remains in effect until either the entry is triggered or the Signal line crosses above zero.

Once we're in a new position, our next step is to place our stops. For a new long position, we set our initial protective stop at the low of the recent countertrend decline minus one point; if we're short, we set our stop at the high of the recent countertrend rally plus one point. We also enable a money-management stop and a $ risk trailing stop.

In addition to stops, we have two other exit conditions: a volatility expansion and a Signal line crossing of the zero line in the opposite direction of the trade. When long, we'll exit at the previous bar's close minus a multiple of the Average True Range. When short, we'll exit at the previous bar's close plus a multiple of the Average True Range. Alternatively, when long we'll exit on the next open when the Signal line crosses below zero. When we're in a short position, we'll exit on the next open when the Signal line crosses above zero.

Defining Our Trading Rules

In this system, we defined both long and short entries as well as exit orders. We enabled a money-management stop. We also did some setup work to calculate the MACD indicator. The setup, entries, and exits are described next.

Setup

a) Calculate a 4-10-16 MACD indicator.

Long Entries

a) Determine the trend. If the Signal line crossed above the zero line more recently than it crossed below the zero line, the trend is up.

b) Locate the first downward crossing of the Signal line by the MACD line since the Signal line crossed above zero.

c) Buy at the close of the previous bar plus a percentage of the Average True Range.

d) The buy setup is in effect until either the entry is triggered or the Signal line crosses below zero.

Short Entries

a) Determine the trend. If the Signal line crossed below the zero line more recently than it crossed above the zero line, the trend is down.

b) Locate the first upward crossing of the Signal line by the MACD line since the Signal line crossed below zero.

c) Sell short at the close of the previous bar minus a percentage of the Average True Range.

d) The setup to sell short is in effect until either the entry is triggered or the Signal line crosses above zero.

Exit Orders

a) Our initial stop when long will be one point below the low of the recent countertrend decline; when short, our initial stop will be one point above the high of the recent countertrend rally.

b) Exit from a long position at the close of the previous bar minus a percentage of the Average True Range.

c) Alternatively, exit on the next open when the Signal line crosses below zero.

d) Exit from a short position at the close of the previous bar plus a percentage of the Average True Range.

e) Alternatively, exit on the next open when the Signal line crosses above zero.

f) We'll also enable a money-management stop and a $ risk trailing stop.

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: First Pullback (STAD6: FirstPullback)

```
Inputs: FastMA(4), SlowMA(10), AvgMA(16), ATRLen(10), EATRPcnt(1.25), XATRPcnt(2.0);
Vars: MACDLine(0), SignalLine(0), ZeroLine(0), ATR(0), UpTrend(False), DnTrend(False),
BuySetup(False), SellSetup(False), CTrendLow(0), CTrendHigh(0), SignalFlag(False);

{Calculation variables}
MACDLine = MACD(Close, FastMA, SlowMA);
SignalLine = XAverage(MACD(Close, FastMA, SlowMA), AvgMA);
```

```
ATR = AvgTrueRange(ATRLen);

{Introduction of an Uptrend and Reset of the Downtrend}
IF SignalLine Crosses Above ZeroLine Then Begin
    UpTrend = True;
    DnTrend = False;
    SellSetup = False;
    SignalFlag = False;
End;

{Introduction of a Downtrend and Reset of the Uptrend}
IF SignalLine Crosses Below ZeroLine Then Begin
    DnTrend = True;
    UpTrend = False;
    BuySetup = False;
    SignalFlag = False;
End;

{Buy Setup}
IF UpTrend Then Begin
    IF MACDLine Crosses Below SignalLine AND SignalFlag = False Then Begin
        BuySetup = True;
        CTrendLow = Low;
    End;
    IF MACDLine < SignalLine AND Low < CTrendLow Then
        CTrendLow = Low;
End;

{Sell Setup}
IF DnTrend Then Begin
    IF MACDLine Crosses Above SignalLine AND SignalFlag = False Then Begin
        SellSetup = True;
        CTrendHigh = High;
    End;
    IF MACDLine > SignalLine AND High > CTrendHigh Then
        CTrendHigh = High;
End;

{Reset of setups when a position is initiated}
IF MarketPosition = 1 Then Begin
    BuySetup = False;
    SignalFlag = True;
End;
IF MarketPosition = -1 Then Begin
    SellSetup = False;
    SignalFlag = True;
End;

{Entries}
IF BuySetup Then
    Buy Next Bar at Close + (EATRPcnt * ATR) Stop;
IF SellSetup Then
    Sell Next Bar at Close - (EATRPcnt * ATR) Stop;
```

```
{Long Exits}
ExitLong Next Bar at CTrendLow - 1 Point Stop;
ExitLong Next Bar at Close - (XATRPcnt * ATR) Stop;

{Short Exits}
ExitShort Next Bar at CTrendHigh + 1 Point Stop;
ExitShort Next Bar at Close + (XATRPcnt * ATR) Stop;

{Additional Exits}
IF DnTrend Then
        ExitLong This Bar on Close;
IF UpTrend Then
        ExitShort This Bar on Close;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
FastMA	4	Price value upon which the Momentum calculation is based.
SlowMA	10	Length, expressed in bars, used to calculate the Momentum of the Price.
AvgMA	16	Length, expressed in bars, used to calculate the Exponential Moving Average of the Momentum.
ATRLen	10	Length, expressed in bars, used to calculate the Average True Range.
EATRPcnt	1.25	The Percentage of the Average True Range that will be added to/subtracted from the base entry price.
XATRPcnt	2.0	The number of bars for which the entry setup is valid.

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

```
Vars: MACDLine(0), SignalLine(0), ZeroLine(0), ATR(0), UpTrend(False), DnTrend(False),
BuySetup(False), SellSetup(False), CTrendLow(0), CTrendHigh(0), SignalFlag(False);
```

Setup

We calculate the MACD based on the Input values and assign it to the variable MACDLine. The signal line is also calculated and assigned to the variable SignalLine. Finally, the Average True Range, which will be used as a part of both the Entries and the Exits, is assigned to the variable ATR.

```
MACDLine = MACD(Close, FastMA, SlowMA);
SignalLine = XAverage(MACD(Close, FastMA, SlowMA), AvgMA);
ATR = AvgTrueRange(ATRLen);
```

The System has several steps in the setup. The first step for the buy setup is to identify the overall trend. A valid up trend occurs when the SignalLine crosses above 0, which is represented by the variable ZeroLine. When an up trend occurs, the UpTrend variable is assigned a value of True. In order to cancel any remaining short-side criteria, DnTrend, SellSetup, and SignalFlag are all set to False.

```
IF SignalLine Crosses Above ZeroLine Then Begin
        UpTrend = True;
        DnTrend = False;
        SellSetup = False;
        SignalFlag = False;
End;
```

The first step for the sell setup is to identify the overall trend. A valid downtrend occurs when the SignalLine crosses below 0, which is represented by the variable ZeroLine. When a downtrend occurs, the DnTrend variable is assigned a value of True. In order to cancel any remaining long-side criteria, UpTrend, BuySetup, and SignalFlag are all set to False.

```
IF SignalLine Crosses Below ZeroLine Then Begin
        DnTrend = True;
        UpTrend = False;
        BuySetup = False;
        SignalFlag = False;
End;
```

If UpTrend is True, indicating the existence of an up trend, we begin the second and final setup stage for the long side. For a BuySetup to be established, the MACDLine must cross below the SignalLine and the SignalFlag must be False. Note that SignalFlag will be set as True once a long entry has occurred. If both criteria are met, BuySetup is assigned a value of True, and the Low is set as the countertrend decline variable CtrendLow. Since the lowest Low of the countertrend decline serves as a protective Stop, if the MACDLine is still below the SignalLine, and the current Low is below the previously defined CTrendLow value, CTrendLow is assigned the value of the new lower low.

```
IF UpTrend Then Begin
        IF MACDLine Crosses Below SignalLine AND SignalFlag = False Then Begin
                BuySetup = True;
                CTrendLow = Low;
        End;
        IF MACDLine < SignalLine AND Low < CTrendLow Then
                CTrendLow = Low;
End;
```

If DnTrend is True, indicating the existence of a downtrend, we begin the second and final setup stage for the short side. For a SellSetup to be established, the MACDLine must cross above the SignalLine and the SignalFlag must be False. Note that SignalFlag will be set as True once a long entry has occurred. If both criteria are met, SellSetup is assigned a value of True, and the High is set as the countertrend rally variable CtrendHigh. Since highest High of the countertrend rally serves as a protective Stop, if the MACDLine is still above the SignalLine, and the current High is above the previously defined CTrendHigh value, CTrendHigh is assigned the value of the new higher High.

```
IF DnTrend Then Begin
    IF MACDLine Crosses Above SignalLine AND SignalFlag = False Then Begin
        SellSetup = True;
        CTrendHigh = High;
    End;
    IF MACDLine > SignalLine AND High > CTrendHigh Then
        CTrendHigh = High;
End;
```

For the Buy Setup, once a long position has been taken, as indicated by the MarketPosition function, the BuySetup variable is set to False. This prevents any additional entries from being generated as a result of that setup. In addition, the SignalFlag variable is set to True, indicating that a position has been taken, as a result of the Buy Setup.

```
IF MarketPosition = 1 Then Begin
    BuySetup = False;
    SignalFlag = True;
End;
```

For the Sell Setup, once a short position has been taken, as indicated by the MarketPosition function, the SellSetup variable is set to False. This prevents any additional entries from being generated as a result of that setup. In addition, the SignalFlag variable is set to True, indicating that a position has been taken, as a result of the Sell Setup.

```
IF MarketPosition = -1 Then Begin
    SellSetup = False;
    SignalFlag = True;
End;
```

Entries

The System entries are fairly straightforward. While the BuySetup is True, a Buy stop order is placed at the Close of the current bar, plus a percentage of the Average True Range. While the SellSetup is True, a Sell stop order is placed at the Close of the current bar, minus a percentage of the Average True Range.

```
IF BuySetup Then
    Buy Next Bar at Close + (EATRPcnt * ATR) Stop;
IF SellSetup Then
    Sell Next Bar at Close - (EATRPcnt * ATR) Stop;
```

Exits

The System has three potential exits built in for each side. The first two are Long Exit stop orders. The first is based on the Low of the countertrend decline, minus 1 point. The second is based on the Close of the current bar, minus a percentage of the Average True Range. Both orders serve as protective stops.

```
ExitLong Next Bar at CTrendLow - 1 Point Stop;
ExitLong Next Bar at Close - (XATRPcnt * ATR) Stop;
```

The next two Exit stop orders are for the short side. The first is based on the High of the countertrend rally, plus 1 point. The second is based on the Close of the current bar, plus a percentage of the Average True Range. Both orders serve as protective stops.

```
ExitShort Next Bar at CTrendHigh + 1 Point Stop;
ExitShort Next Bar at Close + (XATRPcnt * ATR) Stop;
```

The final Exits are Close orders that take effect immediately upon the detection of a change in the overall trend. If DnTrend is True, indicating that the SignalLine has crossed below the ZeroLine, a Long Exit is placed on the Close of the current bar. If UpTrend is True, indicating that the SignalLine has crossed above the ZeroLine, a Short Exit is placed on the Close of the current bar.

```
IF DnTrend Then
        ExitLong This Bar on Close;
IF UpTrend Then
        ExitShort This Bar on Close;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option Do not allow multiple entries in the same direction. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We tested this system on British Pound daily data from 06/93 to 11/98. [Figure 1 - BP daily chart] The system earned $6,925 on 22 trades, with 36% profitable. [Figure 2 - Performance Summary] The largest winner ($4,300) was almost ten times as big as the largest loser (-$437). The average winner ($1,550) far exceeded the average loser (-$391), resulting in a ratio of average win to average loss of 3.96. The average trade (win and loss) netted $314. Our system let profits run for an average of eight bars, but cut losses short in an average of only one bar. The profit factor (2.26) indicates that the system earned $2.26 for each $1.00 it lost.

We also tested this system on Union Carbide weekly data from 01/70 to 11/98, for the long side only. [Figure 3 - UK weekly chart] Our tests showed that the optimal parameters for UK during this time period were a 3-11-12 MACD, a ten-bar Average True Range, a value of .75 for the entry Average True Range, and a value of 1 for the exit Average True Range. A $700 money-management stop and a $900 risk trailing stop tested best. The system earned $3,474 per 100 shares traded. [Figure 4 - Performance Summary] Of the 22 trades, 64% were profitable. The largest winning trade was $1,100 compared to the largest losing trade of $225. The average winning trade was more than three times as large as the average losing trade. The average trade earned $157, and the profit factor was 5.27.

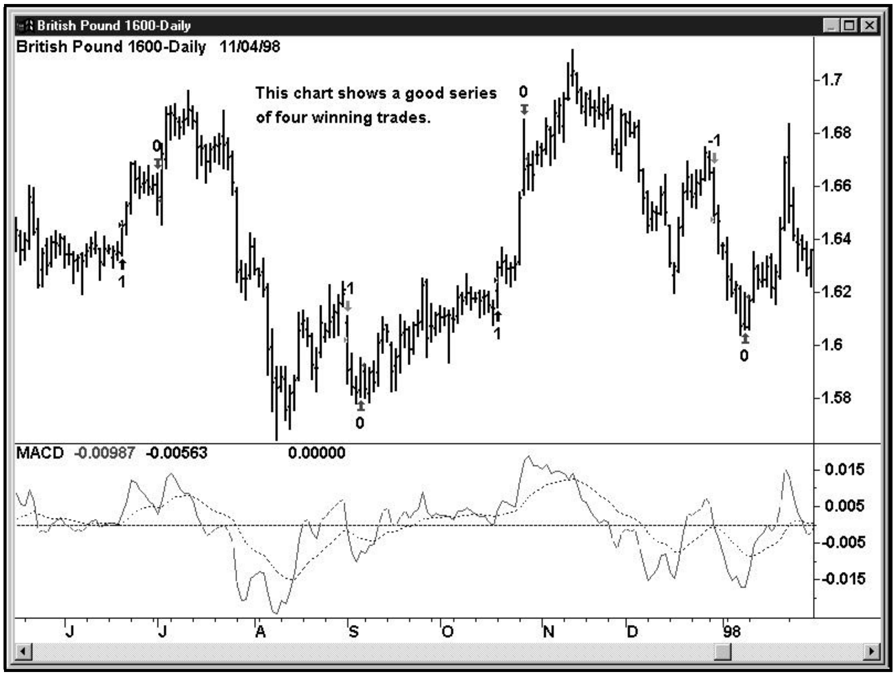
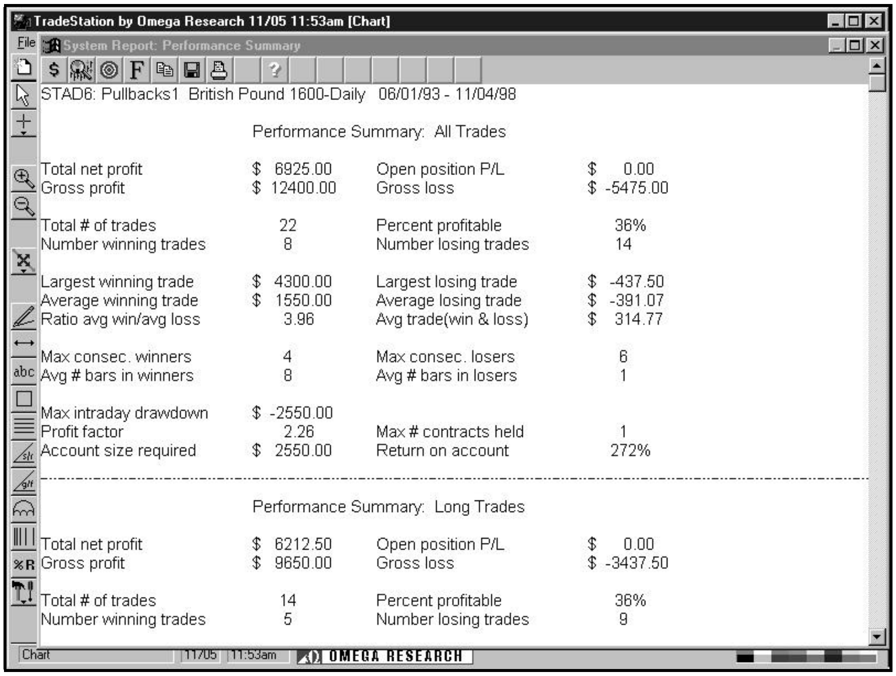


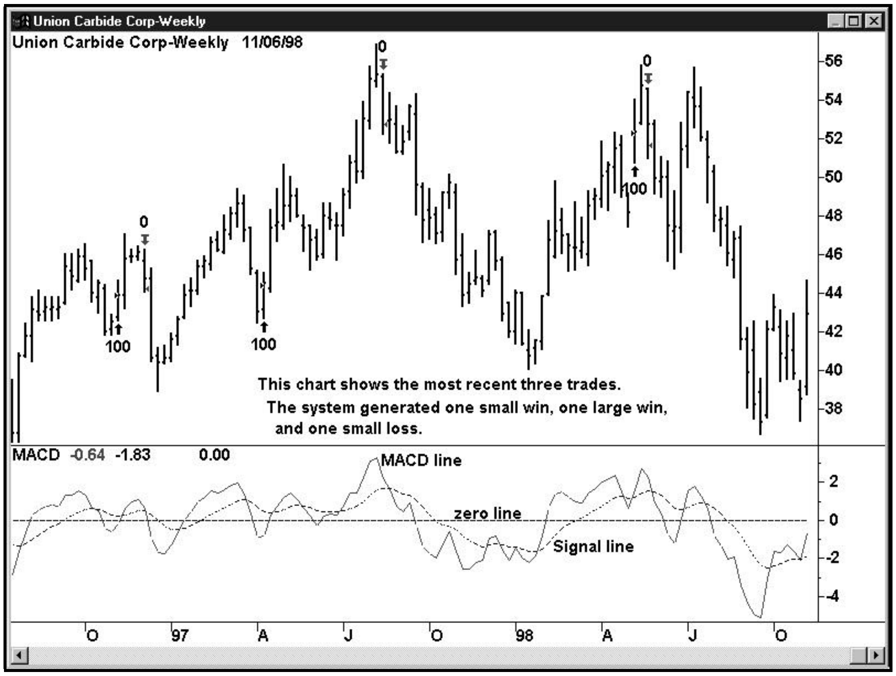
Figure 1 - British Pound Daily Chart



TradeStation by Omega Research 11/05 11:53am [Chart]
File
System Report: Performance Summary
Chart
11/05
11:53am
OMEGA RESEARCH

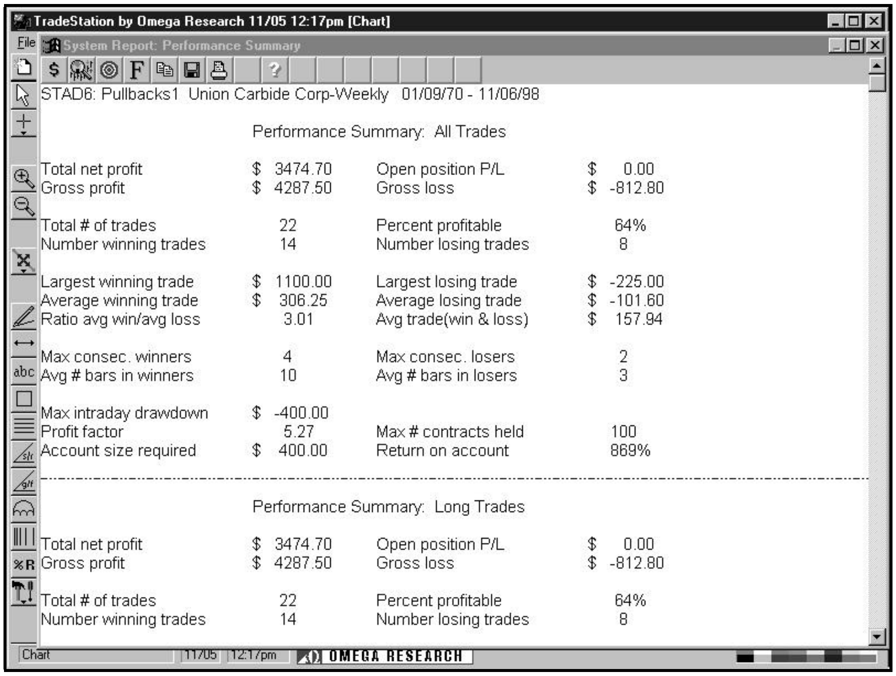
```
STAD6: Pullbacks1  British Pound 1600-Daily  06/01/93 - 11/04/98

                         Performance Summary:  All Trades

Total net profit          $  6925.00    Open position P/L        $     0.00
Gross profit              $ 12400.00    Gross loss               $ -5475.00

Total # of trades             22        Percent profitable          36%
Number winning trades          8        Number losing trades        14

Largest winning trade     $  4300.00    Largest losing trade     $  -437.50
Average winning trade     $  1550.00    Average losing trade     $  -391.07
Ratio avg win/avg loss        3.96      Avg trade(win & loss)    $   314.77

Max consec. winners            4        Max consec. losers           6
Avg # bars in winners          8        Avg # bars in losers         1

Max intraday drawdown     $ -2550.00
Profit factor                 2.26      Max # contracts held         1
Account size required     $  2550.00    Return on account          272%
-------------------------------------------------------------------------------
                         Performance Summary:  Long Trades

Total net profit          $  6212.50    Open position P/L        $     0.00
Gross profit              $  9650.00    Gross loss               $ -3437.50

Total # of trades             14        Percent profitable          36%
Number winning trades          5        Number losing trades         9
```

Figure 2 - British Pound Performance Summary



This chart shows the most recent three trades.
The system generated one small win, one large win, and one small loss.

Figure 3 - Union Carbide Weekly Chart


TradeStation by Omega Research 11/05 12:17pm [Chart]
File
System Report: Performance Summary
$
F
?
abc
%R
Chart
11/05
12:17pm
OMEGA RESEARCH

```
STAD6: Pullbacks1  Union Carbide Corp-Weekly  01/09/70 - 11/06/98

                         Performance Summary:  All Trades

Total net profit          $  3474.70     Open position P/L         $     0.00
Gross profit              $  4287.50     Gross loss                $  -812.80

Total # of trades              22        Percent profitable            64%
Number winning trades          14        Number losing trades          8

Largest winning trade     $  1100.00     Largest losing trade      $  -225.00
Average winning trade     $   306.25     Average losing trade      $  -101.60
Ratio avg win/avg loss          3.01     Avg trade(win & loss)     $   157.94

Max consec. winners            4         Max consec. losers            2
Avg # bars in winners          10        Avg # bars in losers          3

Max intraday drawdown     $  -400.00
Profit factor                   5.27     Max # contracts held          100
Account size required     $   400.00     Return on account             869%
-------------------------------------------------------------------------------

                         Performance Summary:  Long Trades

Total net profit          $  3474.70     Open position P/L         $     0.00
Gross profit              $  4287.50     Gross loss                $  -812.80

Total # of trades              22        Percent profitable            64%
Number winning trades          14        Number losing trades          8
```

Figure 4 - Union Carbide Performance Summary

Suggestions For Improvement

One possible way to improve this system would be to add a "trendiness" indicator such as ADX or the DMI spread. Instead of only trading the first pullback of a trend, we would consider entering new positions subsequent to the first one if ADX was above 20 and rising, or if the DMI spread was above +15 (for a long position) or below -15 (for a short position). This would increase the number of trading opportunities, while ensuring that the trend was still strong.

Parabolic & Volume System

The Parabolic Time/Price System is another idea that Welles Wilder introduced in his book New Concepts in Technical Trading Systems. Parabolic was designed as a stop-and-reverse system, which means that a trader using Parabolic would always have either a long or short position in the market. When Parabolic generates a buy signal, for example, a rising series of dots appears below current prices. As the market moves higher, the dots rise also, first slowly and then more rapidly. When the trend stalls or begins to reverse, the dots and prices meet, the long position is closed out, and a new short position is initiated.

For this system, we wanted to find out if adding a volume requirement to Parabolic entries would improve Parabolic's performance. Our setup to buy is a high reaching the Parabolic dot above the market with volume greater than a five-bar simple moving average of volume. Both conditions must be true on the same bar. The entry for a long position is at the close of the setup bar plus 30% of the distance between the "old" Parabolic (the last dot in the most recent series above the market) and the "new" Parabolic (the first dot in the new series below the market). The entry setup is in effect for four bars. When a long position is entered, the Parabolic series below the market serves as the trailing stop.

The setup to sell short is a low reaching the Parabolic below the market with volume greater than its five bar simple moving average. Both conditions must be true on the same bar. The entry for a short position is at the close of the setup bar minus 30% of the distance between the "old" Parabolic (below the market) and the "new" Parabolic (above the market). The entry setup is in effect for four bars. When a short position is entered, the Parabolic series above the market serves as the trailing stop.

Defining Our Trading Rules

In this system, we defined both long and short entries as well as exit orders. We also did some setup work to calculate the Parabolic and the simple moving average of volume. The setup, entries, and exits are described next.

Setup

a) Calculate the Parabolic.

b) Calculate the five-bar simple moving average of volume.

Long Entries

a) The setup to buy is a high reaching the Parabolic dot above the market with volume greater than its five-bar simple moving average.

b) The long entry is at the close of the setup bar plus 30% of the distance between the "old" Parabolic above the market and the "new" Parabolic below the market.

c) If the long entry is not triggered within four bars of the setup bar, the setup is cancelled.

Short Entries

a) The setup to sell short is the low reaching the Parabolic dot below the market with volume greater than its five-bar simple moving average.

b) The short entry is at the close of the setup bar minus 30% of the distance between the "old" Parabolic below the market and the "new" Parabolic above the market.

c) If the short entry is not triggered within four bars of the setup bar, the setup is cancelled.

Exit Orders

a) For a long position, the exit is a decline to the Parabolic.

b) For a short position, the exit is a rally to the Parabolic.

c) For a long position, the Parabolic series below the market serves as the trailing stop.

d) For a short position, the Parabolic series above the market serves as the trailing stop.

e) We'll also enable a money-management stop.

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: Parabolic & Volume (STAD6: Parabolic/Vol)

```
Inputs: AccFact(.03), VolAvg(5), PcntDiff(.30), SetupLen(4);
Vars: P(0), VAvg(0), BuySetup(False), SellSetup(False), BSFlag(False), BSPrice(0), BSP(0),
SSFlag(False), SSPrice(0), SSP(0);

{Calculation variables}
P = Parabolic(AccFact)[1];
VAvg = Average(Volume, VolAvg);

{setup variables}
BuySetup = High Crosses Above P AND Volume > VAvg;
SellSetup = Low Crosses Below P AND Volume > VAvg;

{Setup}
IF BuySetup Then Begin
        BSFlag = True;
        BSPrice = Close;
        BSP = P[1];
End;
IF SellSetup Then Begin
        SSFlag = True;
        SSPrice = Close;
        SSP = P[1];
End;

{Long Entry}
IF MRO(BuySetup, SetupLen, 1) <> -1 AND BSFlag Then Begin
        value1 = P - BSP;
        Buy Next Bar at BSPrice + (PcntDiff * value1) Stop;
End;

{Short Entry}
IF MRO(SellSetup, SetupLen, 1) <> -1 AND SSFlag Then Begin
        value1 = SSP - P;
        Sell Next Bar at SSPrice - (PcntDiff * value1) Stop;
End;

{Exits}
IF MarketPosition = 1 Then Begin
        ExitLong Next Bar at P Stop;
        BSFlag = False;
End;
IF MarketPosition = -1 Then Begin
        ExitShort Next Bar at P Stop;
        SSFlag = False;
End;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
AccFact	.03	Numeric value that specifies the acceleration factor for the Parabolic calculation.
VolAvg	5	Length, expressed in bars, used to calculate the Moving Average of the Volume
PcntDiff	.30	Numeric value for the percentage of the parabolic calculation used in the entries. (.20 = 20%)
SetupLen	4	The number of bars for which the entry setup is valid.

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

```
Vars: P(0), VAvg(0), BuySetup(False), SellSetup(False), BSFlag(False), BSPrice(0), BSP(0),
SSFlag(False), SSPrice(0), SSP(0);
```

Setup

We begin by assigning the Parabolic and Average Volume calculations to the variables P and VAvg, respectively. Notice that the Parabolic value that is assigned to P is that of 1 bar ago. This is a standard offset of the calculation when using Parabolic.

```
P = Parabolic(AccFact)[1];
VAvg = Average(Volume, VolAvg);
```

There are two conditions required for a buy setup. The High must cross above the Parabolic value and the Volume must be greater than the Average Volume. When both criteria occur on the same bar, BuySetup returns True. The two conditions for a sell setup are that the Low must cross below the Parabolic value and the Volume must be greater than the Average Volume. When both criteria occur on the same bar, SellSetup returns True.

```
BuySetup = High Crosses Above P AND Volume > VAvg;
SellSetup = Low Crosses Below P AND Volume > VAvg;
```

When there is a valid buy setup, three things occur. First, the buy setup flag (BSFlag) is set equal to True. Once a long position is taken, this flag will be set to False, thus preventing any additional entries from that setup. Next, the Close value is assigned to the variable BSFlag for use in the long entry order. Finally, the value of the Parabolic for the previous bar is assigned to the variable BSP. This value will also be used in the long entry order. Notice that in the assignment of the Parabolic value, we are again using an offset of 1. Essentially, we are obtaining an offset value of an already offset variable. We do this in order to obtain the difference between the Parabolic value for the two most recent bars.

```
IF BuySetup Then Begin
	BSFlag = True;
	BSPrice = Close;
	BSP = P[1];
End;
```

When there is a valid sell setup, three things also occur. First, the sell setup flag (SSFlag) is set equal to True. Once a short position is taken, this flag will be set to False, thus preventing any additional entries from that setup. Next, the Close value is assigned to the variable SSFlag, for use in the short entry order. Finally, the value of the Parabolic for the previous bar is assigned to the variable BSP. This value will also be used in the short entry order. Notice that in the assignment of the Parabolic value, we are again using an offset of 1. Essentially, we are obtaining an offset value of an already offset variable. We do this in order to obtain the difference between the Parabolic value for the two most recent bars.

```
IF SellSetup Then Begin
        SSFlag = True;
        SSPrice = Close;
        SSP = P[1];
End;
```

Long Entry

In order for a long entry order to be placed, a True BuySetup must have occurred within SetupLen bars. The MRO function is used to determine if there has been an occurrence of a valid buy setup within SetupLen bars. In addition to the above criterion, the BSFlag must be True, indicating that there has not already been a long entry associated with that setup. The calculation assigned to value1 is simply the difference between the current bar's Parabolic value and that of the previous bar. If the necessary criteria are True, a Long Entry stop order is placed at the BSPrice, plus a percentage of the Parabolic difference specified by value1.

```
IF MRO(BuySetup, SetupLen, 1) <> -1 AND BSFlag Then Begin
        value1 = P - BSP;
        Buy Next Bar at BSPrice + (PcntDiff * value1) Stop;
End;
```

Short Entry

In order for a short entry order to be placed, a True SellSetup must have occurred within SetupLen bars. The MRO function is used to determine if there has been an occurrence of a valid sell setup within SetupLen bars. In addition to the above criterion, the SSFlag must be True, indicating that there has not already been a short entry associated with that setup. The calculation assigned to value1 is simply the difference between the current bar's Parabolic value and that of the previous bar. If the necessary criteria are True, a Short Entry stop order is placed at the SSPrice, minus a percentage of the Parabolic difference specified by value1.

```
IF MRO(SellSetup, SetupLen, 1) <> -1 AND SSFlag Then Begin
        value1 = SSP - P;
        Sell Next Bar at SSPrice - (PcntDiff * value1) Stop;
End;
```

Exits

If the current position is Long, indicated by a MarketPosition equal to 1, a Long Exit Stop order is placed at the value of the Parabolic. In addition, the BSFlag variable is set equal to False so that there are no more orders triggered from that buy setup. If the current position is Short, indicated by a MarketPosition equal to -1, a Short Exit Stop order is placed at the value of the Parabolic. In addition, the SSFlag variable is set equal to False so that there can be no more orders triggered from that sell setup.

```
IF MarketPosition = 1 Then Begin
        ExitLong Next Bar at P Stop;
        BSFlag = False;
End;
IF MarketPosition = -1 Then Begin
        ExitShort Next Bar at P Stop;
        SSFlag = False;
End;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from 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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

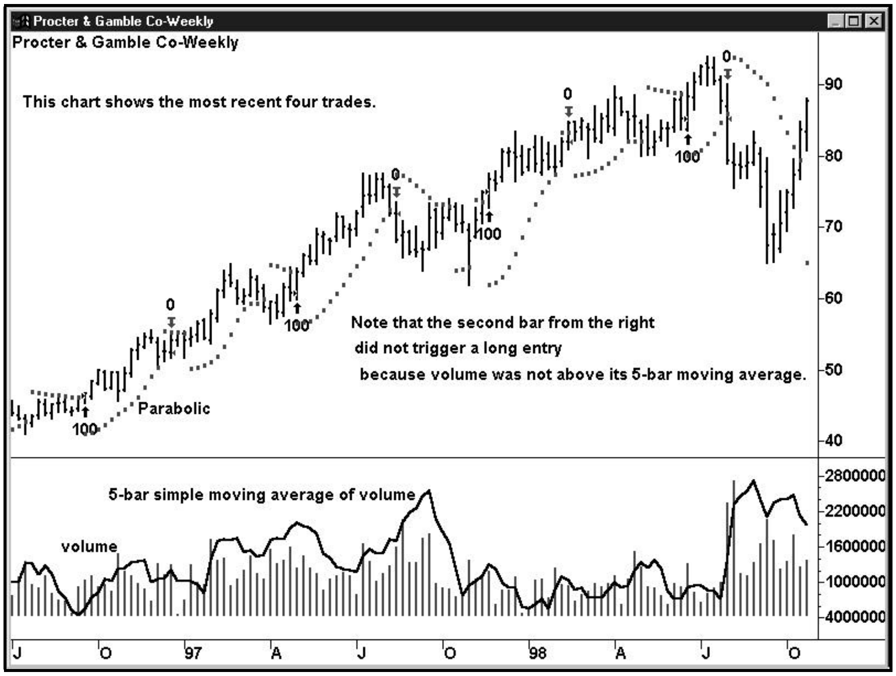
We applied the Parabolic & Volume System to weekly Proctor & Gamble data from 01/78 to 10/98, for the long side only. [Figure 5 - P&G weekly chart] On a position size of 100 shares, the system made $4,411 on 36 trades, with 58% profitable. [Figure 6 - P&G Performance Summary] The largest winning trade ($1,125) was 4.7 times as big as the largest losing trade (-$237.50). The average winner ($266.11) compared to the average loser (-$78.44) yielded a ratio of 3.39 to 1. The system averaged 15 bars in winning trades versus nine bars in losing trades. Perhaps the most impressive number from this performance summary is the profit factor of 4.75 ($4.75 won for every $1.00 lost).

The equity curve rose slowly for the first 16 trades, rose steadily for the next 15 trades, and rose sharply for the remainder of the test period. [Figure 7 - Equity Curve graph]

We also tested this system on daily Swiss Franc data from 06/93 to 10/98. [Figure 8 - SF daily chart] The best-performing parameters were .01 for the acceleration factor, 15 for the average of volume, 50% (.50) for the percent difference between "old" and "new" Parabolic added to or subtracted from the close of the setup bar, and 7 for the number of bars the setup remains in effect.

Applied to the Swiss Franc, the system made $35,187 on 67 trades. [Figure 9 - SF Performance Summary] Only 36% of the trades were profitable, but the average winner was 3.52 times as big as the average loser. The average trade (wins and losses) was $525. Winning trades ran for 24 bars, while losing trades were cut short in only five bars.

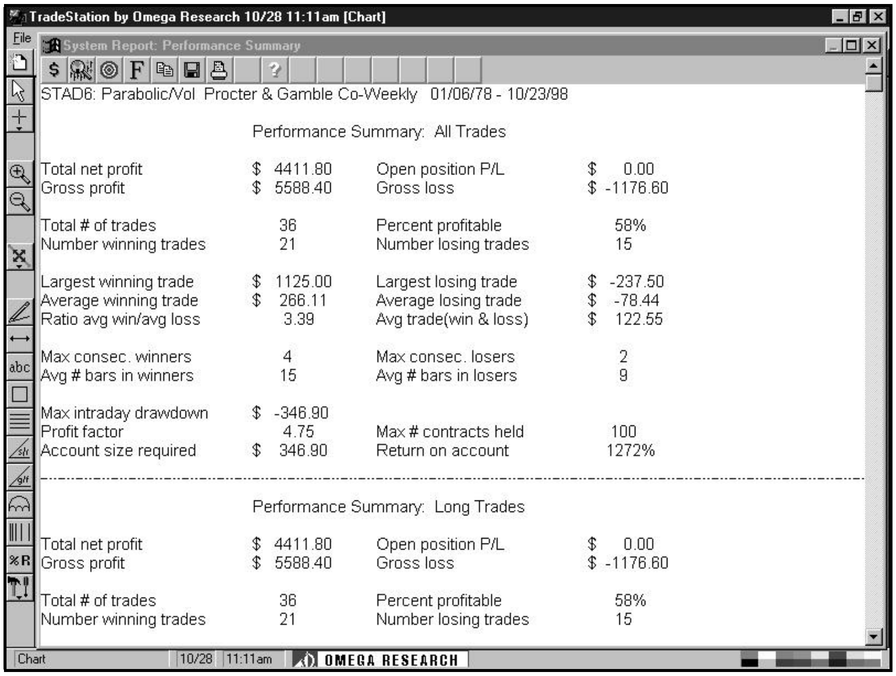
The Total Trades graph shows lots of trades being stopped out at the money-management stop, and two trades qualifying as possible outliers (trades greater than three standard deviations from the average trade). [Figure 10 - Total Trades graph]



This chart shows the most recent four trades.

Note that the second bar from the right did not trigger a long entry because volume was not above its 5-bar moving average.

Figure 5 - Procter & Gamble Weekly Chart


TradeStation by Omega Research 10/28 11:11am [Chart]
File
System Report: Performance Summary
$
F
?
abc
%R
Chart
10/28
11:11am
OMEGA RESEARCH

```
STAD6: Parabolic/Vol  Procter & Gamble Co-Weekly   01/06/78 - 10/23/98

                         Performance Summary:  All Trades

Total net profit          $  4411.80    Open position P/L         $     0.00
Gross profit              $  5588.40    Gross loss                $ -1176.60

Total # of trades              36       Percent profitable            58%
Number winning trades          21       Number losing trades          15

Largest winning trade     $  1125.00    Largest losing trade      $  -237.50
Average winning trade     $   266.11    Average losing trade      $   -78.44
Ratio avg win/avg loss          3.39    Avg trade(win & loss)     $   122.55

Max consec. winners             4       Max consec. losers             2
Avg # bars in winners          15       Avg # bars in losers           9

Max intraday drawdown     $  -346.90
Profit factor                   4.75    Max # contracts held         100
Account size required     $   346.90    Return on account           1272%
------------------------------------------------------------------------------

                         Performance Summary:  Long Trades

Total net profit          $  4411.80    Open position P/L         $     0.00
Gross profit              $  5588.40    Gross loss                $ -1176.60

Total # of trades              36       Percent profitable            58%
Number winning trades          21       Number losing trades          15
```

Figure 6 - Procter & Gamble Performance Summary

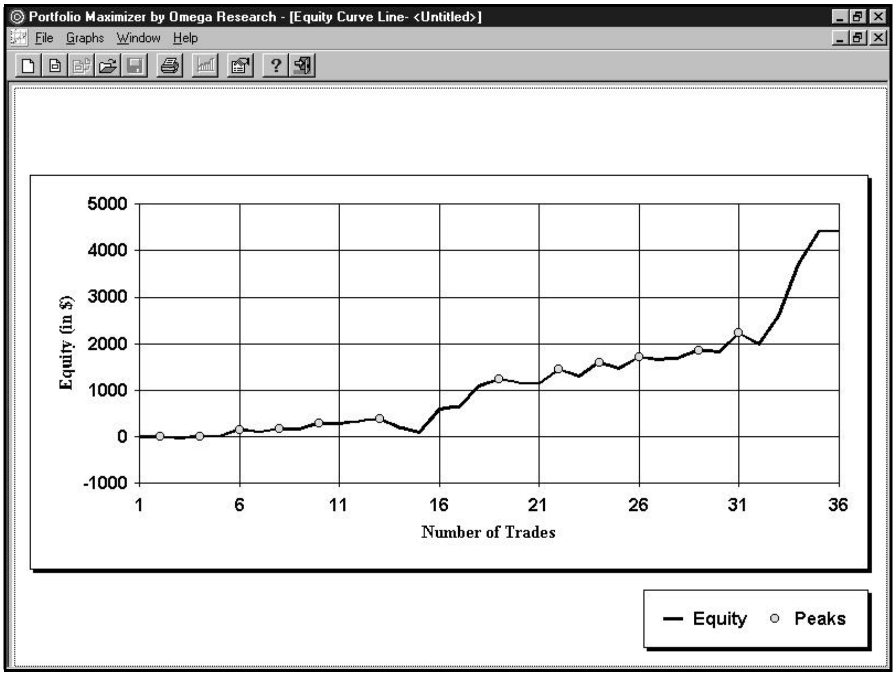


Figure 7 - Procter & Gamble Equity Curve Graph

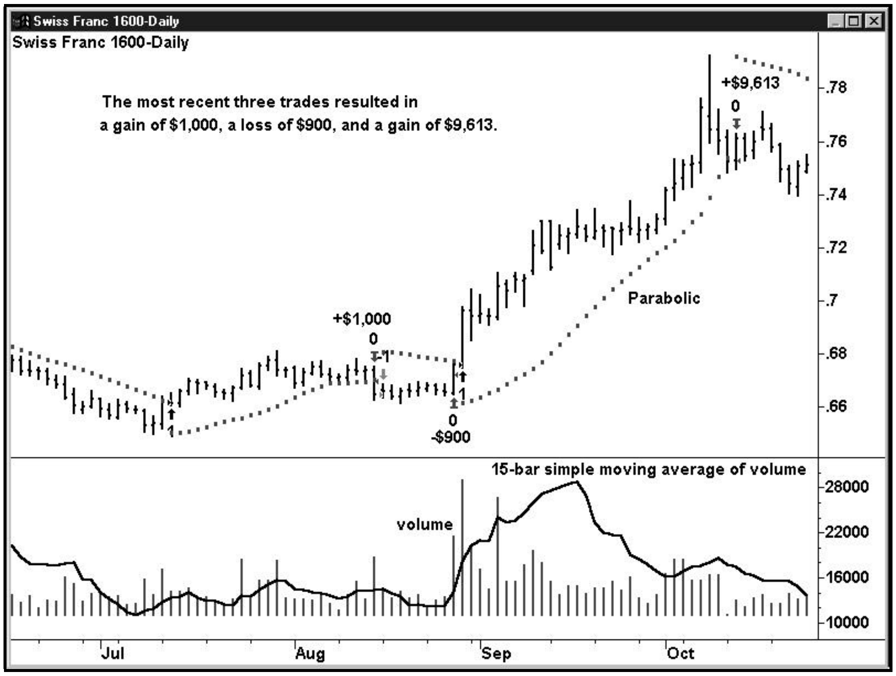
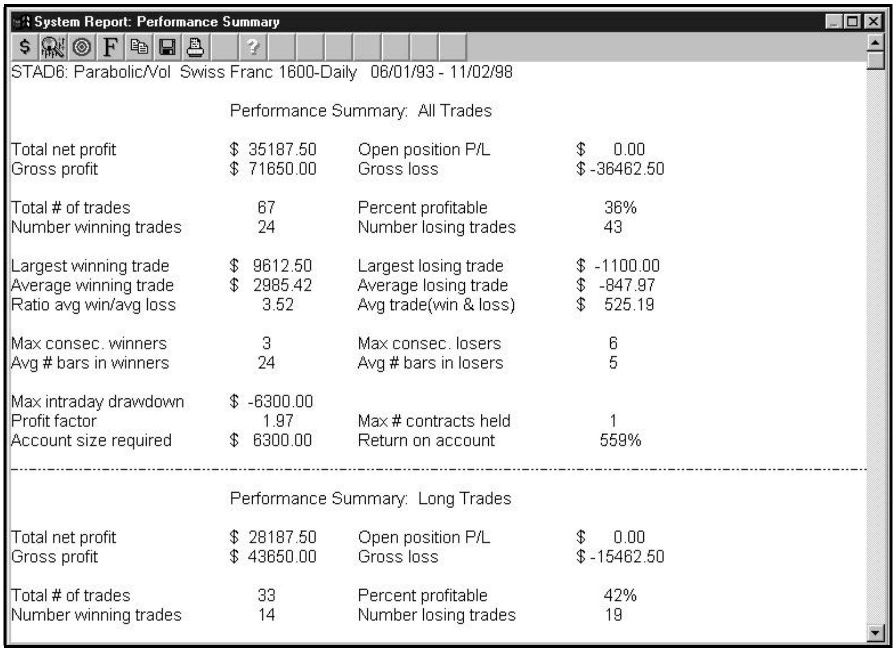


Figure 8 - Swiss Franc Daily Chart



```
STAD6: Parabolic/Vol  Swiss Franc 1600-Daily   06/01/93 - 11/02/98

                          Performance Summary:  All Trades

Total net profit          $  35187.50    Open position P/L         $      0.00
Gross profit              $  71650.00    Gross loss                $ -36462.50

Total # of trades              67        Percent profitable            36%
Number winning trades          24        Number losing trades          43

Largest winning trade     $   9612.50    Largest losing trade      $  -1100.00
Average winning trade     $   2985.42    Average losing trade      $   -847.97
Ratio avg win/avg loss          3.52     Avg trade(win & loss)     $    525.19

Max consec. winners             3        Max consec. losers             6
Avg # bars in winners          24        Avg # bars in losers           5

Max intraday drawdown     $  -6300.00
Profit factor                   1.97     Max # contracts held           1
Account size required     $   6300.00    Return on account            559%
-------------------------------------------------------------------------------
                          Performance Summary:  Long Trades

Total net profit          $  28187.50    Open position P/L         $      0.00
Gross profit              $  43650.00    Gross loss                $ -15462.50

Total # of trades              33        Percent profitable            42%
Number winning trades          14        Number losing trades          19
```

Figure 9 - Swiss Franc Performance Summary

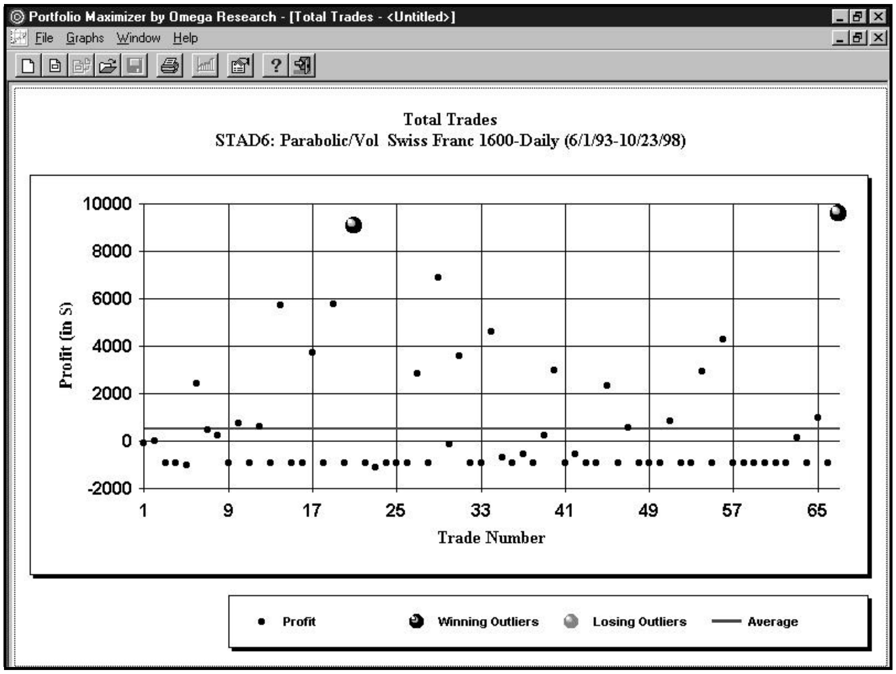


Figure 10 - Swiss Franc Total Trades Graph

Suggestions For Improvement

We liked the performance of Range Leaders in this volume's Displaced Moving-Average Channel & Range Leaders System. What effect would it have on this Parabolic & Volume System to require that the setup bar or one bar immediately before it or after it be a Range Leader?

Volume-Weighted Momentum System

Our Volume-Weighted Momentum System is an attempt to improve the Momentum study by integrating volume into its calculation.

The Momentum oscillator compares today's closing price to the closing price a specified number of days in the past. To calculate a nine-day momentum line, for example, subtract the close of nine days ago from today's close. If you want to create either a faster of a slower oscillator, simply decrease or increase the number of days in the calculation. The formula for the momentum oscillator is: M = C - Cn, where C is the most recent closing price and Cn is the closing price n days ago.

Assuming n = 9, when the nine-day momentum oscillator is above its zero line and rising, nine-day price changes are positive and increasing - that is, the trend is bullish and accelerating. If the momentum line turns flat, it implies that nine-day price changes are about equal during the period of sideways movement. When the momentum oscillator begins to decline from above zero, the market's gains during the past nine days are less than the corresponding gains in the preceding days - that is, the uptrend is decelerating.

When the nine-day momentum oscillator falls below its zero line, the current close is below the close nine days ago. As the downtrend gains bearish velocity (i.e. larger nine-day declines), the momentum line accelerates downward from the zero line. An upturn of the oscillator in negative territory means that the magnitude of nine-day declines is decreasing - that is, the downtrend is decelerating.

The momentum oscillator is a leading indicator - it levels off while prices are still rising in an uptrend or falling in a downtrend, and it reverses its direction when the trend begins to slow down. Since trends generally show a decrease in momentum before reversing their direction, the momentum oscillator can provide and early warning that a change in trend may be imminent.*

The second component of this system is volume - the number of shares or contracts traded in a specified period of time. In an uptrend, volume should increase on the up days and decrease on the down days. In a downtrend volume should increase on the down days and decrease on the up days.

In our Volume-Weighted Momentum System, we want to find out if "synergizing" volume and momentum will create a whole that is greater than the sum of its parts. In other words, volume should improve the performance of the momentum indicator, and momentum should improve the reliability of volume as a tool for analyzing price action.

Our Volume-Weighted Momentum Indicator (VWMI) is constructed as follows: XAverage(Volume*Momentum(close,4),22).

The formula produces a 22-bar exponential moving average of a four-bar momentum indicator that is weighted according to volume. The setup to buy is the VWMI crossing above zero. The buy entry is on the close of the setup bar plus 50% of the five-bar Average True Range. If the entry is not triggered within four bars, the setup is cancelled. The setup to sell short is the VWMI crossing below zero. The entry is on the close of the setup bar minus 50% of the five-bar Average True Range. The setup is cancelled if the entry is not triggered within four bars.

For a long position, the initial stop is the low of the setup bar minus 50% of the five-bar Average True Range. For a short position, the initial stop is the high of the setup bar plus 50% of the five-bar Average True Range.

The exit when long is on the next open after VWMI crosses below zero. The exit when short is on the next open after VWMI crosses above zero. We also will enable a money-management stop and a $ risk trailing stop.

Defining Our Trading Rules

In this system, we defined both long and short entries, as well as exit orders. We also did some setup work to construct the Volume-Weighted Momentum Indicator and to calculate the five-bar Average True Range. The setup, entries, and exits are described next.

Setup

a) Construct the VWMI.

b) Calculate ATR:5.

Long Entries

a) The setup is the VWMI crosses above zero.

b) The entry is at the close of the setup bar plus 50% of ATR:5.

c) The buy setup will remain in effect for four bars after the setup bar.

Short Entries

a) The setup is the VWMI crosses below zero.

b) The entry is at the close of the setup bar minus 50% of ATR:5.

c) The sell setup will remain in effect for four bars after the setup bar.

Exit Orders

a) Our initial stop when long will be the low of the setup bar minus 50% of ATR:5.

b) Our initial stop when short will be the high of the setup bar plus 50% of ATR:5.

c) Exit a long position on the next open when VWMI crosses below zero.

d) Exit a short position on the next open when VWMI crosses above zero.

e) We'll also enable a money-management stop and a $ risk trailing stop.

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: Volume Weighted Momentum (STAD6: Vol Momentum)

```
Inputs: Price(Close), MomLen(5), AvgLen(20), ATRLen(5), ATRPcnt(.25), SetupLen(5);
Vars: VWM(0), ATR(0), LEPrice(0), SEPrice(0), BullSetup(False), BearSetup(False), LSetup(0),
SSetup(0);

{Calculation variables}
VWM = XAverage(Volume * Momentum(Price, MomLen), AvgLen);
ATR = AvgTrueRange(ATRLen);

{Setup variables}
BullSetup = VWM Crosses Above 0;
BearSetup = VWM Crosses Below 0;

{Establishment of Buy Setup}
IF BullSetup Then Begin
        LSetup = 0;
        LEPrice = Close;
End;

{Establishment of Sell Setup}
IF BearSetup Then Begin
        SSetup = 0;
        SEPrice = Close;
End;

{Setup Accumulators}
LSetup = LSetup + 1;
SSetup = SSetup + 1;

{Entries}
IF LSetup <= SetupLen Then Begin
        Buy Next Bar at LEPrice + (ATRPcnt * ATR) Stop;
        ExitLong Next Bar at Low - (ATRPcnt * ATRLen) Stop;
End;
IF SSetup <= SetupLen Then Begin
        Sell Next Bar at SEPrice - (ATRPcnt * ATR) Stop;
        ExitShort Next Bar at High + (ATRPcnt * ATRLen) Stop;
End;

{Exits}
IF MarketPosition = 1 Then Begin
        LSetup = SetupLen;
        IF BearSetup Then
                ExitLong Next Bar at Market;
End;
IF MarketPosition = -1 Then Begin
        SSetup = SetupLen;
        IF BullSetup Then
                ExitShort Next Bar at Market;
End;
```

Inputs

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

INPUT	DEFAULT	DESCRIPTION
Price	Close	Price value upon which the Momentum calculation is based.
MomLen	5	Length, expressed in bars, used to calculate the Momentum of the Price.
AvgLen	20	Length, expressed in bars, used to calculate the Exponential Moving Average of the Momentum.
ATRLen	5	Length, expressed in bars, used to calculate the Average True Range.
ATRPcnt	.25	The Percentage of the Average True Range that will be added to/subtracted from the base entry price.
SetupLen	5	The number of bars for which the entry setup is valid.

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

```
Vars: VWM(0), ATR(0), LEPrice(0), SEPrice(0), BullSetup(False), BearSetup(False), LSetup(0),
SSetup(0);
```

Setup

The value for the Exponential Moving Average of the Momentum is assigned to the variable VWM. The value for the Average True Range calculation is assigned to the variable ATR.

```
VWM = XAverage(Volume * Momentum(Price, MomLen), AvgLen);
ATR = AvgTrueRange(ATRLen);
```

For a bullish setup, the Volume Weighted Momentum must cross above 0. When this occurs, the BullSetup variable will return True. For a bearish setup, the Volume Weighted Momentum must cross below 0. When this occurs, the BearSetup variable will return True.

```
BullSetup = VWM Crosses Above 0;
BearSetup = VWM Crosses Below 0;
```

When a bullish setup occurs, two things happen. First, the LSetup variable that counts the number of bars in the setup is set to 0. Second, the Close is assigned to the variable LEPrice, which will serve as the basis for the Long Entry order.

```
IF BullSetup Then Begin
        LSetup = 0;
        LEPrice = Close;
End;
```

When a bearish setup occurs, two things happen as well. First, the SSetup variable that counts the number of bars in the setup is set to 0. Second, the Close is assigned to the variable SEPrice, which will serve as the basis for the Short Entry order.

```
IF BearSetup Then Begin
	SSetup = 0;
	SEPrice = Close;
End;
```

LSetup and SSetup are incremented by 1 on each bar in order to keep count of the bars since the beginning of the setup. Remember that the counters below are reset each time there is a new setup.

```
LSetup = LSetup + 1;
SSetup = SSetup + 1;
```

Long Entry

If the LSetup is less than SetupLen, meaning that we are in the midst of a valid bullish setup period, a Long Entry and Initial Exit order are triggered. The Long Entry Stop order is placed at the LEPrice, plus a percentage of the Average True Range value. The initial Long Exit Stop order is placed at the Low, minus a percentage of the Average True Range value.

```
IF LSetup <= SetupLen Then Begin
	Buy Next Bar at LEPrice + (ATRPcnt * ATR) Stop;
	ExitLong Next Bar at Low - (ATRPcnt * ATRLen) Stop;
End;
```

Short Entry

If the SSetup is less than SetupLen, meaning that we are in the midst of a valid bearish setup period, a Short Entry and Initial Exit order are triggered. The Short Entry Stop order is placed at the SEPrice, minus a percentage of the Average True Range value. The initial Short Exit Stop order is placed at the High, plus a percentage of the Average True Range value.

```
IF SSetup <= SetupLen Then Begin
	Sell Next Bar at SEPrice - (ATRPcnt * ATR) Stop;
	ExitShort Next Bar at High + (ATRPcnt * ATRLen) Stop;
End;
```

Long Exit

When the current position is Long, expressed by MarketPosition equal to 1, the setup is nullified by setting the LSetup variable equal to the SetupLen value, thus preventing any additional entries as a result of the setup. When the Volume Weighted Momentum crosses below 0, essentially the same criterion as for a BearSetup, an Long Exit order is placed at the open of the next bar (market).

```
IF MarketPosition = 1 Then Begin
	LSetup = SetupLen;
	IF BearSetup Then
		ExitLong Next Bar at Market;
End;
```

Short Exit

When the current position is Short, expressed by MarketPosition equal to -1, the setup is nullified by setting the SSetup variable equal to the SetupLen value, thus preventing any additional entries as a result of the setup. When the Volume Weighted Momentum crosses above 0, essentially the same criterion as for a BullSetup, a Short Exit order is placed at the open of the next bar (market).

```
IF MarketPosition = -1 Then Begin
     SSetup = SetupLen;
     IF BullSetup Then
            ExitShort Next Bar at Market;
End;
```

General System Format

When we apply a system to a chart, we can use the options in the Format dialog box to format costs, stops, and properties. We did not enter an amount for slippage and commission although those costs must certainly be taken into account before a system is traded.

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

Under the Stops tab, we enabled a money-management stop. The money-management stop holds the dollar amount per position or per contract/share we want to risk before exiting from the position. We also enabled a $ risk trailing stop.

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 will type the maximum number of dollars you are willing to risk per contract traded.*

In the Properties tab, we selected the option **Do not allow multiple entries in the same direction**. If the system is in a position, and market conditions generate another entry order in the same direction, the order is ignored.

Testing & Improving

We tested this system on daily data of Intel from 03/78 to 10/78, on the long side only. [Figure 11 - Intel daily chart] On 62 trades, the system earned $9,790 (per 100 shares). [Figure 12 - Intel Performance Summary] Fifty percent of the trades were winners with an exceptional ratio of average win to average loss of 5.88. The largest winner ($3,893) was much greater than the largest loser (-$537), and the average trade made $157.

Winning trades ran for an average of 48 bars, while losing trades were cut short after an average of 14 bars. The only area in which the system didn't perform well was maximum consecutive winners (5) versus maximum consecutive losers (12).

We also tested the VWMI system on daily data of the US Dollar Index from 06/93 to 10/98. [Figure 13 - US $ Index daily chart] The parameters that tested best for this market were five for the momentum length, 20 for the moving average length, five for the Average True Range length, one for the number of Average True Range, and six for the setup length. The total net profit was $22,320 per contract on 70 trades of which 50% were profitable. [Figure 14 - US $ Index Performance Summary] The largest winner ($6,680) was much bigger than the largest loser (-$1,400), and the average winner of $1,256 was more than twice the size of the average loser of -$619. Winners ran for an average of 19 bars, while losses were cut short on an average of eight bars.

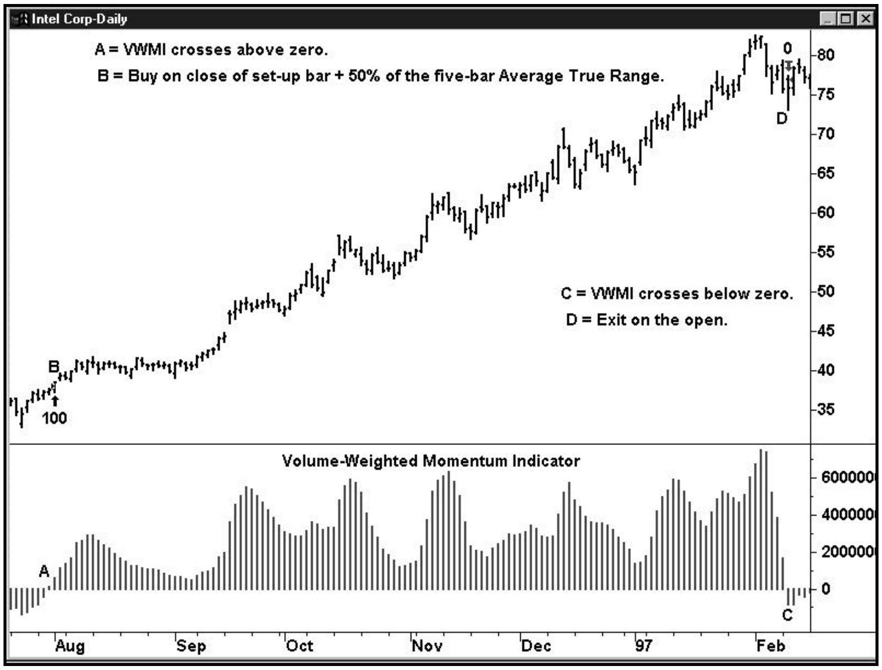
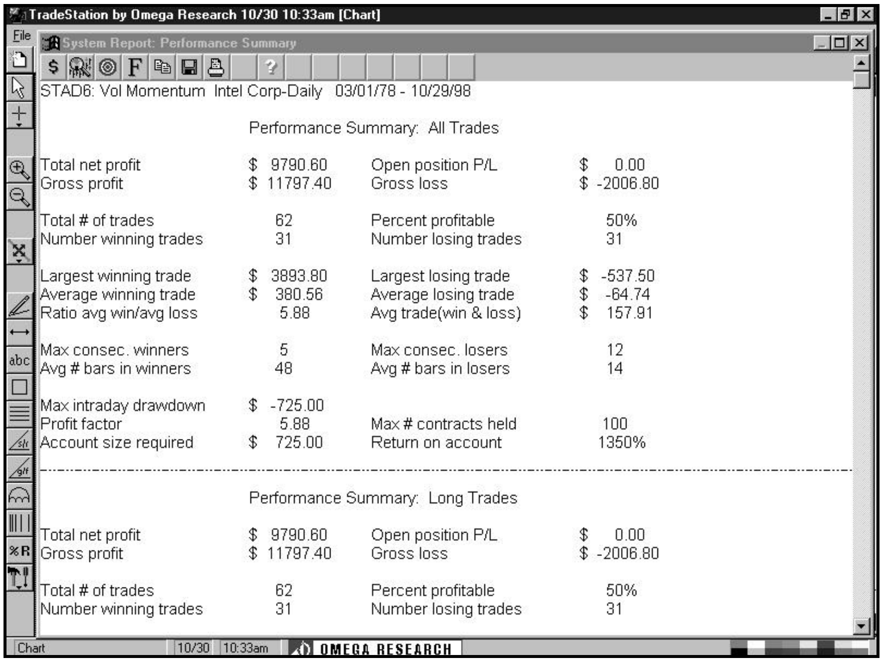
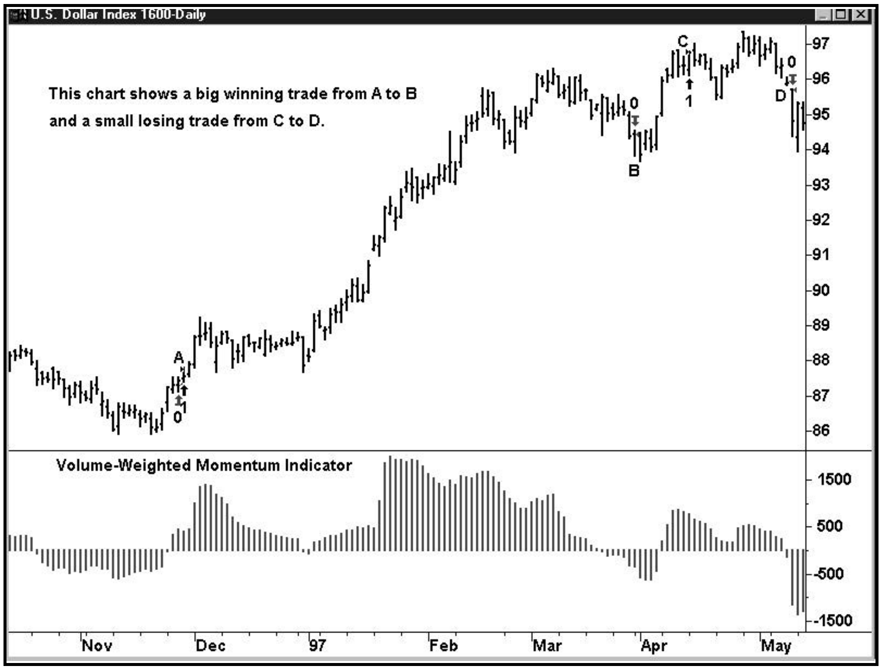


Figure 11 - Intel Daily Chart


TradeStation by Omega Research 10/30 10:33am [Chart]
File
System Report: Performance Summary
Chart
10/30
10:33am
OMEGA RESEARCH

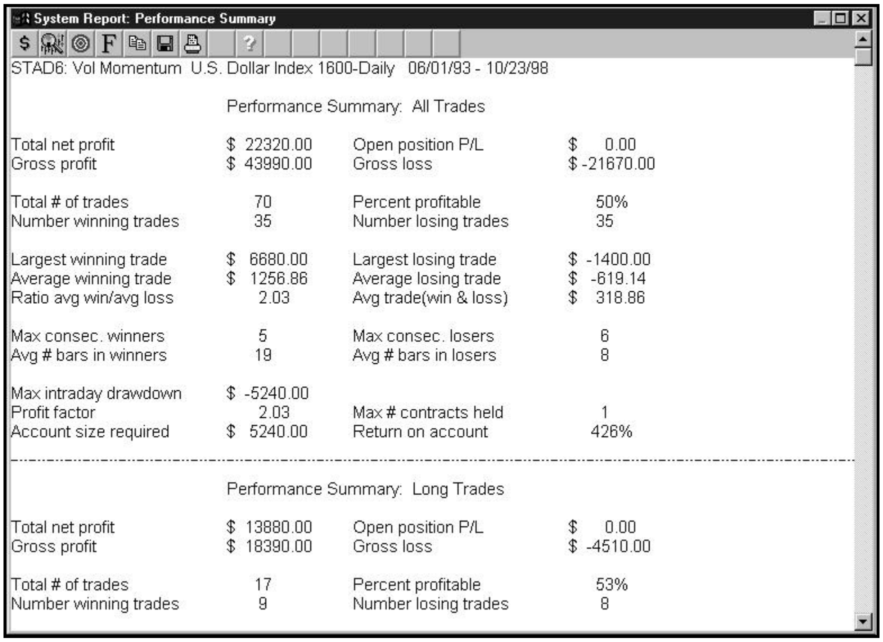
```
STAD6: Vol Momentum  Intel Corp-Daily   03/01/78 - 10/29/98

                              Performance Summary:  All Trades

Total net profit          $  9790.60     Open position P/L        $     0.00
Gross profit              $ 11797.40     Gross loss               $ -2006.80

Total # of trades             62         Percent profitable           50%
Number winning trades         31         Number losing trades         31

Largest winning trade     $  3893.80     Largest losing trade     $  -537.50
Average winning trade     $   380.56     Average losing trade     $   -64.74
Ratio avg win/avg loss         5.88      Avg trade(win & loss)    $   157.91

Max consec. winners            5         Max consec. losers           12
Avg # bars in winners         48         Avg # bars in losers         14

Max intraday drawdown     $  -725.00
Profit factor                  5.88      Max # contracts held        100
Account size required     $   725.00     Return on account          1350%
-----------------------------------------------------------------------------

                              Performance Summary:  Long Trades

Total net profit          $  9790.60     Open position P/L        $     0.00
Gross profit              $ 11797.40     Gross loss               $ -2006.80

Total # of trades             62         Percent profitable           50%
Number winning trades         31         Number losing trades         31
```

Figure 12 - Intel Performance Summary



This chart shows a big winning trade from A to B and a small losing trade from C to D.

Figure 13 - US $ Index Daily Chart



```
STAD6: Vol Momentum  U.S. Dollar Index 1600-Daily   06/01/93 - 10/23/98

                        Performance Summary:  All Trades

Total net profit          $  22320.00     Open position P/L         $     0.00
Gross profit              $  43990.00     Gross loss                $ -21670.00

Total # of trades             70          Percent profitable            50%
Number winning trades         35          Number losing trades          35

Largest winning trade     $   6680.00     Largest losing trade      $  -1400.00
Average winning trade     $   1256.86     Average losing trade      $   -619.14
Ratio avg win/avg loss         2.03       Avg trade(win & loss)     $    318.86

Max consec. winners            5          Max consec. losers             6
Avg # bars in winners         19          Avg # bars in losers           8

Max intraday drawdown     $  -5240.00
Profit factor                  2.03       Max # contracts held           1
Account size required     $   5240.00     Return on account            426%
--------------------------------------------------------------------------------

                        Performance Summary:  Long Trades

Total net profit          $  13880.00     Open position P/L         $     0.00
Gross profit              $  18390.00     Gross loss                $  -4510.00

Total # of trades             17          Percent profitable            53%
Number winning trades          9          Number losing trades           8
```

Figure 14 - US $ Index Performance Summary

Suggestions For Improvement

When we designed this system, we didn't consider whether the zero line was the best value for entry or exit crossover. The standard assumption for many indicators is that above zero is bullish, and below zero is bearish. That may not be correct in this case. Perhaps the system could be improved by finding the optimal values for long entry, short entry, long exit, and short exit instead of only considering the zero line.

CHAPTER 5:

Learning More About System Trading & Development: Universalization — A Modified Optimization Approach

In the April, 1993, issue of Technical Analysis of Stocks & Commodities magazine, Adam White wrote an excellent article called "Filtering Breakouts." One topic White introduced in the article was Universalization, which he described as "a modified optimization approach."

At Omega Research, we believe strongly in the value of optimization. We believe that optimization, when applied properly, can be very useful in improving trading systems. One point we emphasize every chance we get is that optimization should be used to fine-tune trading systems that are already performing well, not to create trading systems. Optimization should be one of the last steps in the development of a trading system, not one of the first steps.

We think that White's Universalization idea may prove helpful to many traders who want to use the power of TradeStation to optimize their systems, but who are also wary of the dangers of over-optimization.

Following is an example of the universalization process applied to our Channel Break Intrabar system. First, select the markets you want to trade with your system. For this example, we chose U.S. Bonds, British Pound, Cotton, Crude Oil, Deutsche Mark, Eurodollars, Value Line Index, and Wheat. Second, run standard optimizations for the value you want to improve. We chose to optimize the channel length of our Channel Break Intrabar system. Third, rank the optimized values from largest to smallest. Here's our example:

Market	**Optimized Channel Length**
Crude Oil	47
Wheat	45
Cotton	32
U.S. Bonds	29
British Pound	27
Value Line Index	25
D-Mark	21
Eurodollars	13

Fourth, drop the largest and the smallest values. In our example, drop 47 and 13. Fifth, average the remaining values. In our example, we add 45, 32, 29, 27, 25, and 21 for a total of 179 and an average of 29.8, which we round up to 30.

The Universalized parameter is 30. Thirty wasn't the top-performing value for any of the eight markets in our example, but the universalization process strongly suggests that it is a reasonable, ball-park number for trading this system. Applied to our portfolio of eight markets, the 30-bar channel breakout may not turn out to be the best possible choice (with the benefit of hindsight), but it is very unlikely to rank among the worst choices.

The following table shows the results of three tests: (1) a test of the Channel Break Intrabar default value of 10; (2) a test to optimize for the most profitable value; and (3) a test of the universalized value.

Market 01/87-12/96	Ten-Bar Default Net Profit	Optimized Value/ Net Profit	Universalized Value (30) Net Profit
Crude Oil	$10,970	47 / $41,190	$26,630
Wheat	7,031	45 / 17,331	2,481
Cotton	16,570	32 / 53,705	18,375
Bonds	16,688	29 / 59,656	56,344
British Pound	52,063	27 / 87,350	78,425
Value Line	(9,825)	25 / 69,725	2,525
D-Mark	37,988	21 / 45,225	24,088
Eurodollars	12,925	13 / 25,825	12,975

In this example, universalization produced $77,433 more net profit than the default value, gaining $221,843 compared to the default value's $144,410. Note that the universalized value produced more profits than the default value in six of the eight markets we tested. Of course, the optimized values made a lot more money than either the default values or the universalized value; however, we would want to carefully backward and forward test the optimized values to see how well they hold up on out-of-sample data before we would trade with them.

The technique of applying universalization to a portfolio of optimized values seems to have some merit. Universalization is clearly not a replacement for optimization. It is, however, an interesting step that can be added to our optimization routine as we try to make our systems more stable and robust.

APPENDIX A

Volume in Review

Each volume of the System Trading & Development Club provides feedback on previous volumes. Feedback can include general comments from members as well as answers to specific questions regarding one of the systems. This appendix will also address any discrepancies or omissions, if any, made from the systems in previous volumes.

Our Moving-Average Crossover System from Volume One of STAD Club generated the following question from one of our members:

> **Q:** One of the things I noticed is that when I change the Lcount variable from -999 to -2 it doesn't seem to affect anything. Why is this? From what I thought, the -999 was supposed to add one bar and then reset to zero. Is this correct?
>
> **A:** The basic idea of the system is to find when Condition 1 is met, and to perform a certain action the following ten bars. A simplified version of the EasyLanguage instructions used in this system follows:

```
Input:  ChanLen(10);
Vars:  Lcount(-999);
Condition1={Buy Setup Condition};
If Condition 1 then
Lcount=BarNumber
```

The goal behind initializing the variable Lcount with -999 is to avoid false signals in the first few bars of the chart. We are going to be looking for the BarNumber to be less than Lcount plus a certain number of bars (specified by the input ChanLen). So, supposing that Condition1 is NOT met throughout the first ten bars, and we are in bar number 1, then the last IF statement

```
    If BarNumber<Lcount+ChanLen then...
```

evaluates to IF 1 < -2 + 10 then...which is true, even though our condition has not been met yet. By initializing the variable Lcount to -999, this instruction will read IF 1 < -999 + 10 then...which will be false, just as we intended it to be.

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!

INDEX