THE SUPER COMBO DAY TRADING STRATEGY

So far we have concentrated on longer-term trend-following systems using daily bar analysis. We are now going to work with strategies that deal with intraday bars. Working with intraday bars is the same as working with daily bars. TradeStation doesn't care what time frame you are working with (tick, minute, daily, or weekly); all built-in indicators and functions work the same. The only time programming gets complex is when you are dealing with more than one time frame at a time. Many systems trade on intraday bars, but use daily bars to calculate their buy and sell signals. The Super Combo strategy falls under this category. This system is designed to day trade the stock indices. We should call this the kitchen sink day trader, because we have thrown a lot of different ideas into this one system. This system is rather complex and, therefore, so is the programming code. If you can understand this code, then we doubt there are too many trading ideas that you couldn't program. We will program a system in a modular format to make it easier to understand. In addition, the program code will be heavily laden with comments to further aid in digestion.

Of all of the day trade systems that Futures Truth monitors and ranks, the systems that feature both break out and failed break out technology always tops the list of best performers. Instead of reinventing the wheel, we will borrow this overall concept and use it as a foundation for our own system.

First off, let's discuss the break out component of the Super Combo. If you look at a daily bar on almost any trading instrument, you will notice that the high price is always greater than or equal to the open and close. On non-limit days, it is always higher than the low. At some time during the day, there was more demand than there was supply and, therefore, the market moved up above the open. Conversely, when supply was greater than demand, the market moved down. The objective of an open range break out system is to try and capture some of the market movement between the open and high (or close) or the open and low (or close). The key to successful break out systems is finding the "sweet spot" above or below the open that foretells further movement in that direction. The worst feeling in the world for a day trader is buying at or near the high and selling at or near the low. Your buy point must be above the high of the day when the market has no direction and at the same time must be well below the high when the market has a strong directional move. Your sell point must have the same attributes. You are probably thinking to yourself that this is impossible. You would be right. But we all know that trading is not impossible, difficult yes, but not impossible. This is the beauty of the Super Combo; it knows it must stick its neck out and buy/sell break outs, but it also knows that there is a great likelihood that the initial break out will fail. This is where the failed break out logic kicks into high gear. Once a certain price level is achieved (upside or downside), the system switches gears and begins to look

for failed break out opportunities. How many times have you seen a bar chart where the open is near the high and the close is near the low or just the opposite? These are perfect examples of failed break out opportunities. These concepts are how the Super Combo enters the market.

Once a trade has been initiated, the system uses protective, break-even, and trailing stops to manage the position. The system also utilizes a protective stop reversal. Sometimes a trade is put on and failure occurs within the next couple of hours. This can occur without the system sensing a failed break out. So, under certain conditions, we allow the system to reverse its position at the protective stop level, even when a failed break out isn't signaled. After the system reaches a certain profit level, the protective stop moves to a break-even level. Later in the afternoon, when the likelihood of profit taking and retracements of the overall trend increases, the protective or break-even stops switch to a trailing stop. It may seem to you that the exit techniques are much more complex and thought out than the entry techniques. Many traders don't realize that trade management determines the success of the entry technique and, therefore, they spend all their time trying to figure out the best way to get into the market. Research time should be split between entries and exits. If all goes well with our entry, the position will be closed at the end of trading. There is one more important point concerning this system: no trade can be taken within 30 minutes of the open time. This filter (a technique to help pick out good trades) allows the market to digest any reports that come out before and slightly after the stock indices open. If you take a look at the first 30 minutes of a stock index, especially on report days, you will discover an extreme level of volatility. This volatility will play havoc with any break out-based trading strategy. So, let's just skip it.

Now that we know the overall approach of the Super Combo, let's get a little more specific. The open range break out shouldn't be foreign to you because we described it in the Thermostat strategy. Nor should the buy easier day and sell easier day concept be unfamiliar. The Super Combo system utilizes both ideas to calculate the entry points for the break out component. The first step is to see if we can even take a trade today. We have spent many hours researching the consequences of wide range and small range bars and have drawn the conclusion that small range bars usually give a hint to a subsequent wide range bar. So, if yesterday was a small range bar, then we can attempt a trade today; or else we simply skip today and look for something else to do. We consider a bar to be a small range bar (SRB) by comparing the absolute value of the Open – Close of yesterday's bar to the average of the absolute values of the past ten days' Open – Close values. If the bars' Open to Close range is less than 85 percent of the ten-day average open to close range, then it is considered an SRB. Once we find out that we can trade, we must then determine if today is a buy or sell easier day. A buy easier day occurs when the close is less than or equal to the previous day's close. A sell easier day is simply the con-

verse; today's close is greater than the previous day's close. (Note: We give a slight bias to the buy easier day in case the close of today and yesterday are exactly the same.) Remember, we are making it easier to buy or sell; we are not ruling out a potential buy or sell. Prior to the bear market of 2000, you would be surprised at the number of systems that only bought or made it difficult to sell. If today is a buy easier day, we can buy at the open of today plus 30 percent of the ten-day average true range, and sell at the Open of today minus 60 percent of the ten-day average range (don't use true ranges). On the other hand, if today is a sell easier day, we can sell at the open of today minus 30 percent of the ten-day average range, and buy at the open of today plus 60 percent of the ten-day average range.

Now that we understand the break out component of the Super Combo, let's delve into the failed break out methods. In terms of our system, the following three scenarios define a failed upside break out:

1. The market gaps above yesterday's high a certain amount and then trades below it a certain amount.
2. The market opens below and moves above yesterday's high a certain amount and then trades below it a certain amount.
3. A long position is initiated and failure occurs before 12:00 P.M. central time (CT). Failure is defined as being stopped out. We take advantage of this if failure doesn't happen too quickly.

The opposite is true for a failed break out to the downside. We look to get in the market when any of these three scenarios occur. This type of entry can be defined as countertrend; a position is being entered against the prevailing trend.

When we enter a trade from a break out method, we use a protective stop equal to 25 percent of the average range or three full points, whichever is greater. By using a percentage-based protective stop, we are adapting our protection to the current market condition. If a market is exhibiting high volatility, then a tight fixed stop will get you stopped out with a loss on most trades. On the flipside of the coin, if the market is exhibiting low volatility, then a large fixed stop will risk more than the potential reward and eventually net failure. Adaptive parameters or stops must have boundaries so they won't take on ridiculous values, hence the three-point floor. In today's volatile markets, a protective stop of less than three-points is a disaster waiting to happen. We use the same protective stop on the failed break out entry with one exception. If we enter a position at a stop loss level, we use 15 percent instead of 25 percent of the average range or three full points. The reason behind the different protective stop level for this entry is that there is a chance the market may not have any direction for the entire day. The market proved indecisive by stopping our initial position out. We will nibble on what the market is offering, but we will use a tight stop in case the market performs a double whammie. If the market

is looking kindly on us and provides a profit equal to 50 percent of the 10-day average range, then we pull our protective stop up to a break-even point. In doing so, we achieve a free trade minus slippage and commission. Later in the afternoon, specifically after 2:30 P.M. CT, we trail our protective stop below/above the low/high of the prior 3 five minute bars. Again, through extensive studies, we have discovered that the market will more times than not, fade the overall daily trend in the last 30 to 45 minutes of trading. Losing all of the day's profit in the last fifteen minutes is almost as heartbreaking as buying the high and selling the low. Since we are not trying to scalp the market, we will only test the waters twice on a daily basis. In other words, we will only initiate one long position and/or one short position during the day. Once we have entered the market twice, we stop looking for another entry. That's it. Finito. We have just about given you everything except the kitchen sink. There should be enough ideas in this one system to build a whole slew of different strategies.

The Super Combo sounds rather complicated, doesn't it? If you think it sounds complicated, wait until we try to program it. The best way to attack this monster is one modular block at a time. First, let's pseudocode all of the calculations that deal with daily bars. This may be a good time to introduce TradeStation's capabilities that deal with multiple data streams and time frames. With TradeStation and EasyLanguage, you can analyze up to five different data and/or time frames. In the case of the Super Combo, we will deal with two different time frames: five minute and daily. We will use the daily bar data to feed our calculations and the five-minute data to actually enter our trades. Why five-minute bars instead of fifteen-minute or any other time interval bars? you may ask. If at all possible, it is always best to test your trading ideas on the smallest bar increment that you can. If we had the computer power and time, we would have tested on individual tick bars. The higher the time resolution, the more accurate your testing will be. We will let the online TradeStation help further explain this:

> Calculating Orders on Historical Data Time-based bars include the Open, High, Low, and Close for the specified time period. When you work with historical data, TradeStation doesn't know the chronological order of the transactions that make up the bar. The only transactions for which the chronology is known are the Open, which occurred first, and the Close, which occurred last. With time-based bars based on historical data there is no way to know whether the market opened and then went down, or the market opened and then went up.
>
> However, because the order of ticks can be important, two general rules were established about price movement and the chronological order in which ticks occur.
>
> Assumption 1—The order in which prices on a bar are reached relates to the proximity of the Open to the Low and to the High.

When the Open is closer to the Low than to the High, TradeStation assumes that the Low was reached first. Likewise, if the Open is closer to the High, TradeStation assumes that the High was reached first. For example, say you have a buy order at 100. You protect yourself against losses at 97 points or lower, and want to exit the position at 101 or higher. The Open is 99 points. The High of the day is 103 and the Low is 92, so the Open is closer to the High. In this case, your buy order is filled at 100 and your strategy will generate your profit-taking exit order at 101, thus recording a profit of 1 point.

However, it's possible that in reality the price climbed to 100, at which point your entry order was filled, and then it dipped, hitting 99, before climbing back up to 101. In this case, you would actually have taken a loss of 1 point instead of a profit of 1 point. So, looking at historical data, and based on the market assumption described earlier, TradeStation would record this trade as a winner, when it would in fact have been a loser. The opposite could happen also, where a trade could be recorded as a loser when it was actually a winner.

To more closely simulate market activity in these cases, TradeStation developed what is known as Bouncing Ticks. When enabled, Trade-Station automatically sends the data down by a percentage (the TradeStation default is 10%) immediately after a buy or sell order, and then comes back up to the next item on the bar (High, Low, or Close). This action more closely resembles what happens using real-time data.

Using this 10 percent setting on the example above, the buy would have happened at 100, and would have dropped immediately by 10% to 99 (to more closely resemble real-time results), your exit would be recorded at 99, and the bar would more correctly reflect the losing trade that would have happened if you had been using real-time data.

Assumption 2—Symbols trade at every price along the bar.

The second assumption is that a symbol trades at every price along the bar. That may not always be an accurate assumption. If, using the example above again, you had a buy order at 100, and the symbol actually traded at 99 and then jumped to 101, the strategy would record your buy at 100, even though you actually were filled at 101. If your exit was at 110 then, the strategy would record a profit of 10 points, even though your true profit would be only 9 points. This assumption might cause a strategy to appear more or less profitable than it actually was.

TradeStation allows you to avoid both of these assumptions by setting a back-testing resolution. Setting a back-testing resolution allows your strategies to be evaluated according to the actual prices in the order they occurred.

For instructions on enabling the Bouncing Ticks option, see "Setting the Bouncing Ticks Option."

The smaller the time increment that you are dealing with the less there is a chance for error. A fifteen-minute bar is made up of three 5-minute bars. Inside the fifteen-minute bar there may be a price that was never traded; a gap between the high or low and the subsequent open price on a five-minute bar.

Figure 6.5   Incorrect Daily Bar Assumption

Also, the ambiguity of which occurred first, the high price or the low price, decreases with a smaller time frame. Figure 6.5, which is the same as Figure 4.6, again illustrates how the accuracy of actual market movement increases with smaller time interval bars.

Now that we know why we are using five-minute bars, let's move on. Before we got on the back-testing data resolution tangent, we were about to program the daily bar based calculations.

Super Combo Daily Data Bar Calculation Pseudocode

```
averageRange = Average(Range of Daily Data,10)
averageOCRange = Average(Abs(Open-Close of Daily Data),10)
canTrade = Abs(Open-Close of Daily Data) < 85% of averageOCRange
buyEasierDay = 0
sellEasierDay = 0
if(Close of Daily Data <= Close[1] of Daily Data) then buyEasierDay = 1
if(Close of Daily Data > Close[1] of Daily Data) then sellEasierDay = 1
if(buyEasierDay) then
        buyBOPoint = Open of tomorrow + 30% of averageRange
        sellBOPoint = Open of tomorrow – 60% of averageRange
if(sellEasierDay) then
        sellBOPoint = Open of tomorrow – 30% of averageRange
        buyBOPoint = Open of tomorrow + 60% of averageRange
longBreakPt = High of Daily Data + 25% of averageRange {upside break out
                                                 achieved}
shortBreakPt = Low of Daily Data – 25% of averageRange {dnside break out
                                                 achieved}
longFBOPoint = Low of Daily Data + 25% of averageRange {dnside BO failure
                                                 buy pt}
shortFBOPoint = High of Daily Data – 25% of averageRange {upside BO failure
                                                 sell pt}
```

Now for the 5-minute bar pseudocode. We must initialize some variables on the first bar of the day.

```
if(Date <> Date[1]) then     {a little trick to determine the first bar
                              of the day}
      barCount    = 1
      intraHigh   = 0
      intraLow    = 99999999
      buysToday   = 0
      sellsToday  = 0
      currTrdType = 0
```

EasyLanguage does not have a convenient way to keep track of the intraday high and low, so let's just do it ourselves.

```
if(High > intraHigh) intraHigh = High    {must keep high of current date
                                         manually}
if(Low < intraLow) intraLow = Low        {must keep low of current date
                                         manually}
```

If this bar's date is the same as the previous bar, then we know that we are progressing through today.

```
if(Date = Date[1]) then barCount = barCount + 1
```

If barCount is greater than 6, then we know that we have surpassed the first six bars of the day.

```
if(barCount > 6) then         {skipped first 6 – 5min bars or first 30 minutes}
```

Let's keep track of the number of buy entries and sell entries for today.

```
if(MarketPosition = 1) then buysToday = 1    {if we are long we must have bought
                                             today}
if(MarketPosition = -1) then sellsToday = 1 {same goes for being short}
```

Now let's enter the market on a buy or sell break out.

```
if(buysToday = 0 and Time < 1430) then Buy("LBreakOut") next bar at
   buyBOPoint stop
if(sellsToday = 0 and Time < 1430) then Sell("SbreakOut") next bar at
   sellBOPoint stop
```

Has the market exceeded yesterday's High plus 25 percent of the average range?

```
if(intraHigh > longBreakPoint and Time < 1430) then
```

If it has, look to sell as the market moves back down to yesterday's high – 25 percent of the average range.

```
if(sellsToday = 0) then SellShort("SfailedBO") next bar at shortFBOPoint stop
```

Another failed break out indicator occurs when the market breaks out and gets us into a position and then reverses and stops us out. We can enter a short position on this type of failed breakout if we are stopped out from a long position and at least four 5 minute bars have passed since we entered the long position and it is before 12:00 p.m. central time. We don't allow a reversal if the market immediately stops us out – this is usually a sign of a knee jerk reaction.

```
if(Time < 2300 and sellsToday = 0 and
     longLiqPoint <> EntryPrice and BarsSinceEntry >= 4) then
           SellShort("LongLiqRev") next bar at longLiqPoint stop;
```

Ditto for the failed break out on the short side.

```
if(intraLow < shortBreakPoint and Time < 1430) then
      if(buysToday = 0) then Buy("BfailedBO") next bar at longFBOPoint stop

if(Time < 1200 and buysToday = 0 and
      shortLiqPoint <> EntryPrice and BarsSinceEntry >= 4) then
           Buy("ShortLiqRev") next bar at shortLiqPoint stop;
```

Now this is where it gets a little complicated with the trade management algorithm.

```
if(MarketPosition = 1) then
```

Most of the time, our long liquidation point will be the entry price – 25 percent of the average range or three full points whichever is greater.

```
      longLiqPoint = MinList(EntryPrice – 25% of averageRange,EntryPrice - 3.00
      points) {normal reversal use 25%}
```

However, if we are reversing a short position at a protective stop level then we must use 15 percent. EasyLanguage doesn't have a simple way to determine which filter got you into the current trade. We can figure if the previous position was short and the current position is long and the last bar was the trigger bar and the high of the trigger bar is greater than or equal to our short liquidation point, *and* our short liquidation point is closer to the market than the short failed break out point, then we know we were reversed on to long from a short liquidation point.

```
if(MarketPosition(1) = -1 and BarsSinceEntry(0) = 1 and
High[1]>=shortLiqPoint and shortLiqPoint < shortFBOPoint) then
curTrdType  = -2
if(curTrdType = -2) then
        longLiqPoint = MinList(EntryPrice – 15% of
        averageRange,EntryPrice - 3.00 points)  {long liq reversal use 15%}
```

If we are long and the high of the five-minute bar has exceeded the Entry Price plus 50 percent, then we need to move our stop to break even.

```
if(High > EntryPrice + 50% of averageRange) then longLiqPoint = EntryPrice
```

If time is greater than 2:30 P.M. CT, then we start trailing our long liquidation point from the lowest low of the past 3 five-minute bars.

```
if(time >= 1430) then longLiqPoint = MaxList(lonqLiqPoint,Lowest(Low,3))
```

Order entry module for either liquidating our long position or initiating a short position at our long liquidation point.

```
if(Time < 1200 and sellsToday = 0 and longLiqPoint <> EntryPrice) and
BarsSinceEntry >= 4) then
        Sell Short ("LongLiqRev") next bar at longLiqPoint stop
else
        Sell ("LongLiq") next bar at longLiqPoint stop
```

Ditto for the short side.

```
If(MarketPosition = -1) then
        shortLiqPoint = MaxList(EntryPrice + 25% of averageRange,EntryPrice +
          3.00 points) {normal reversal use 25%}
        if(MarketPosition(1) = 1 and BarsSinceEntry(0) = 1 and
        (Low[1] <= longLiqPoint and longLiqPoint > longFBOPoint) then
        curTrdType  = +2
        if(curTrdType = +2) then
                shortLiqPoint = MinList(EntryPrice – 15% of
                averageRange,EntryPrice - 3.00 points)
                {long liq reversal use 15%}
        if(Low <= EntryPrice – 50% of averageRange) shortLiqPoint = EntryPrice
        if(Time < 1200 and buysToday = 0 and shortLiqPoint <> EntryPrice) then
                Buy ("ShortLiqRev") next bar at shortLiqPoint stop
        else
                BuyToCover ("ShortLiq") next bar at shortLiqPoint stop
        If(time >= 1430) then shortLiqPoint =
        MinList(shortLiqPoint, Highest(High,3))
```

This next function call gets us out of the market at the closing bell.

```
SetExitOnClose
```

Did you notice how our pseudocode evolved into almost pure EasyLanguage? The more programming that you get under your belt, the less English-like you will be with your pseudocode. You may think you can save time by cutting out this middle step, but you can't. John and I have programmed thousands of systems, and simply outlining the structure and variables ahead of time always saves time in the long run.

Super Combo Code

```
{Super Combo by George Pruitt
This intraday trading system will illustrate the multiple data handling
capabilities of TradeStation. All pertinent buy and sell calculations will be
based on daily bars and actual trades will be executed on 5-min bars. I have
made most of the parameters input variables.}

Inputs:waitPeriodMins(30),initTradesEndTime(1430),liqRevEndTime(1200),
thrustPrcnt1(0.30),thrustPrcnt2(0.60),breakOutPrcnt(0.25),
failedBreakOutPrcnt(0.25),protStopPrcnt1(0.25),protStopPrcnt2(0.15),
protStopAmt(3.00),breakEvenPrcnt(0.50),avgRngLength(10),avgOCLength(10);

Variables:averageRange(0),averageOCRange(0),canTrade(0),buyEasierDay(FALSE),
sellEasierDay(FALSE),buyBOPoint(0),sellBOPoint(0),longBreakPt(0),
shortBreakPt(0),longFBOPoint(0),shortFBOPoint(0),barCount(0),
intraHigh(0),intraLow(999999),buysToday(0),sellsToday(0),
currTrdType(0),longLiqPoint(0),shortLiqPoint(0),yesterdayOCRRange(0),
intraTradeHigh(0),intraTradeLow(999999);

{Just like we did in the pseudocode—let's start out with the daily bar
calculations. If Date <> Date[1]—first bar of day}
if(Date <> Date[1]) then {save time by doing these calculations once per day}
begin
    averageRange = Average(Range,10) of Data2; {Data 2 points to daily bars}
    yesterdayOCRRange = AbsValue(Open of Data2-Close of Data2);
    averageOCRange = Average(AbsValue(Open of Data2-Close of Data2),10);
    canTrade = 0;
    if(yesterdayOCRRange< 0.85*averageOCRange) then canTrade = 1;
    buyEasierDay = FALSE;
    sellEasierDay = FALSE;
    {See how we refer to Data2 - the daily data}
    if(Close of Data2 <= Close[1] of Data2) then buyEasierDay = TRUE;
    if(Close of Data2 > Close[1] of Data2) then sellEasierDay = TRUE;

    if(buyEasierDay) then
    begin
         buyBOPoint = Open of data1 + thrustPrcnt1*averageRange;
         sellBOPoint = Open of data1 - thrustPrcnt2*averageRange;
```

```
    end;
    if(sellEasierDay) then
    begin
          sellBOPoint = Open of data1 - thrustPrcnt1*averageRange;
          buyBOPoint = Open of data1 + thrustPrcnt2*averageRange;
    end;

    longBreakPt = High of Data2 + breakOutPrcnt*averageRange;
    shortBreakPt = Low of Data2 - breakOutPrcnt*averageRange;

    shortFBOPoint = High of Data2 - failedBreakOutPrcnt*averageRange;
    longFBOPoint = Low of Data2 + failedBreakOutPrcnt*averageRange;

{Go ahead and initialize any variables that we may need later on in the day}
    barCount = 0;
    intraHigh = 0;intraLow = 999999; {Didn't know you could do this}
    buysToday = 0;sellsToday = 0;{You can put multiple statements on one
      line}
    currTrdType = 0;
end;   {End of the first bar of data}
{Now let's trade and manage on 5-min bars}
if(High > intraHigh) then intraHigh = High;
if(Low < intraLow ) then intraLow = Low;
barCount = barCount + 1; {count the number of bars of intraday data}
if(barCount > waitPeriodMins/BarInterval and canTrade = 1) then
{have we waited long enough—wait PeriodMin is an input variable and
BarInterval is set by TradeStation. Wait PeriodMins = 30 and BarInterval = 5,
so 30/5 = 6}
begin
    if(MarketPosition = 0) then
    begin
          intraTradeHigh = 0;
          intraTradeLow = 999999;
    end;

    if(MarketPosition = 1) then
    begin
          intraTradeHigh = MaxList(intraTradeHigh,High);
          buysToday = 1;
    end;
    if(MarketPosition =-1) then
    begin
          intraTradeLow = MinList(intraTradeLow,Low);
          sellsToday = 1;
    end;

    if(buysToday = 0 and Time < initTradesEndTime) then
          Buy("LBreakOut") next bar at buyBOPoint stop;
    if(sellsToday = 0 and Time < initTradesEndTime) then
```

```
            SellShort("SBreakout") next bar at sellBOPoint stop;
    if(intraHigh > longBreakPt and sellsToday = 0 and Time <
        initTradesEndTime) then
            SellShort("SfailedBO") next bar at shortFBOPoint stop;
    if(intraLow < shortBreakPt and buysToday = 0 and Time <
        initTradesEndTime) then
            Buy("BfailedBO") next bar at longFBOPoint stop;
{The next module keeps track of positions and places protective stops}
    if(MarketPosition = 1) then
    begin
            longLiqPoint = EntryPrice - protStopPrcnt1*averageRange;
            longLiqPoint = MinList(longLiqPoint,EntryPrice - protStopAmt);
            if(MarketPosition(1) = -1 and BarsSinceEntry = 1 and
                High[1] >= shortLiqPoint and shortLiqPoint < shortFBOPoint)
                    then
                currTrdType = -2; {we just got long from a short liq reversal}
            if(currTrdType = -2) then
            begin
                longLiqPoint = EntryPrice - protStopPrcnt2*averageRange;
                longLiqPoint = MinList(longLiqPoint,EntryPrice -
                    protStopAmt);
            end;
            if(intraTradeHigh >= EntryPrice + breakEvenPrcnt*averageRange)
                then
                longLiqPoint = EntryPrice; {BreakEven trade}
            if(Time >= initTradesEndTime) then
                longLiqPoint = MaxList(longLiqPoint,Lowest(Low,3)); {Trailing
                    stop}
            if(Time < liqRevEndTime and sellsToday = 0 and
                longLiqPoint <> EntryPrice and BarsSinceEntry ≥ 4) then
            begin
                SellShort("LongLiqRev") next bar at longLiqPoint stop;
        end
        else begin
            Sell("LongLiq") next bar at longLiqPoint stop;
        end;
    end;
    if(MarketPosition =-1) then
    begin
        shortLiqPoint = EntryPrice+protStopPrcnt1*averageRange;
        shortLiqPoint = MaxList(shortLiqPoint,EntryPrice + protStopAmt);
        if(MarketPosition(1) = 1 and BarsSinceEntry(0) = 1 and
            Low [1] <= longLiqPoint and longLiqPoint > longFBOPoint) then
            currTrdType = +2; {we just got long from a short liq reversal}
        if(currTrdType = +2) then
        begin
            shortLiqPoint = EntryPrice + protStopPrcnt2*averageRange;
            shortLiqPoint = MaxList(shortLiqPoint,EntryPrice + protStopAmt);
        end;
```

```
        if(intraTradeLow <= EntryPrice - breakEvenPrcnt*averageRange) then
            shortLiqPoint = EntryPrice; {BreakEven trade}
        if(Time >= initTradesEndTime) then
            shortLiqPoint = MinList(shortLiqPoint,Highest(High,3));
                {Trailing stop}
        if(Time < liqRevEndTime and buysToday = 0 and
            shortLiqPoint <> EntryPrice and BarsSinceEntry ≥ 4) then
        begin
            Buy("ShortLiqRev") next bar at shortLiqPoint stop;
        end
        else begin
            BuyToCover("ShortLiq") next bar at shortLiqPoint stop;
        end;
    end;
end;
SetExitOnClose;
```

The Super Combo program demonstrates how to:

- Use the keyword *date* to determine the first intraday bar of the day.
- Create and program a complete trade management scheme.
- Use the keyword *time* to determine optimum trading periods.
- Use and manage multiple data input.
- Exit all positions at the close of the day.

Super Combo trading performance is summarized in Tables 6.7 and 6.8. Since we have a limited amount of historical intraday data to work with, the performance statistics found in Table 6.8 were generated with our Excalibur testing software.

A visual example of how this system enters and exits trades is shown in Figure 6.6.

Super Combo Summary

Wow, this strategy was complicated to program. It wouldn't have been if Easy-Language had provided a little bit more information on which filter initiated the current trade. The complicated part of the program revolves around determining if we were reversed on a liquidation point and which protective stop to use. We used one liquidation variable for all of our trade management stops; the value of the liquidation variable changed dependent on trade type, time, and open trade profit. If you have simpler exit schemes, you can use the built-in EasyLanguage functions for protective stops and profit objectives. Analyzing different time frames with multiple data streams is much easier than doing the same with only one data stream. You could have done the same thing with

Table 6.7
Super Combo Performance Table 1

TradeStation Strategy Performance Report—SuperCombo @SP-5 min (2/12/2001–4/18/2002) $100 for Slippage and Commission			
Performance Summary: All Trades			
Total Net Profit	27,175.00	Open position P/L	0.00
Gross Profit	126,725.00	Gross Loss	(99,550.00)
Total # of trades	155	Percent profitable	38.06%
Number winning trades	59	Number losing trades	96
Largest winning trade	7,400.00	Largest losing trade	(2,125.00)
Average winning trade	2,147.88	Average losing trade	(1,036.98)
Ratio avg win/avg loss	2.0713	Avg trade (win & loss)	175.32
Max consec. Winners	5	Max consec. losers	10
Avg # bars in winners	48	Avg # bars in losers	20
Max intraday drawdown	(13,725.00)		
Profit Factor	1.2730	Max # contracts held	1
Account size required	13,725.00	Return on account	198.00%

a single five-minute bar chart, but the programming would be considerably more difficult. You could build daily bars from the intraday bars and keep track of all the daily statistics. This could be accomplished by creating array variables and keeping track of when the day begins and ends and the highest and lowest point achieved on an intraday basis. Like we said previously, most traders don't use arrays in their programming, but they are handy when you start doing advanced analysis. For this reason, we have included a strategy that incorporates arrays in Chapter 8.

Back to the subject of using multiple data streams. We have just found that multiple time frame analysis is easier when you can refer to a different data stream for each time frame. Super Combo did relatively well over the past six years. Remember, we haven't done any work with optimizing the many variables in this system. We leave any optimization, tweaking, or application to other markets to you, the reader. The framework of Super Combo is the same framework of many systems that sell for thousands of dollars, so it is up to you to transform this system into your own and make it ten times better.

Table 6.8
Super Combo Performance Table 2

Super Combo Performance from 1986–2/28/2002

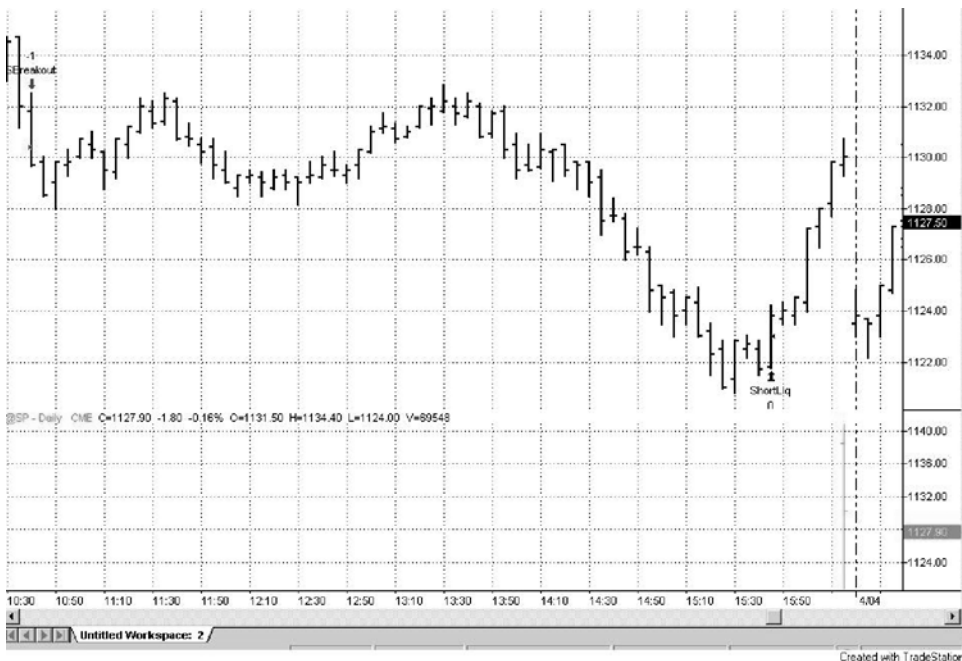| SP500 5 minute bars	Test 16.17 years		4079 trading days
Total Net P/L	188913$	Avg. Net/Year	11685$
Optimal f	0.19	Geometric Mean	1.003
%Winning Months	59%	Avg. %Ret. MaxDD/Year	39%
		Avg. %Ret. with Time	165%
Max DrawDn ClsTrd	24025$		
Max DrawDn w/OTE	24025$		
Best Trade	10350$	Worst Trade	−2750$
Average Trade	89$	Net Prof:Loss Ratio	1.3
Average Win	981$	Average Loss	−568$
Long Net P/L	104288$	Short Net P/L	84625$
# of Trades	2129	Avg. Trades/Year	132
# of Winning Trade	902	Percent Winners	42.40%
# of Losing Trades	1227	Most Cons. Losses	11
Avg. Days Per Trade	0.5	Longest Flat Time	1077 days
%Time in the Market	24%	Sharpe Ratio	0.25

Figure 6.6 Super Combo Trade