

# Day Trading with Zone Trading Analysis

An **EasyLanguage** Framework for Testing Market Activity Among  
User Defined  
Zones

## Introduction

Break out and counter trend trading has been the most popular and successful day trading



mechanisms for many, many years. Basically, it attempts to capture the big trend days via break outs and the counter trend moves via intraday reversals. If the market consistently broke out to either the upside or downside and kept going, then day trading would be like an ATM machine. Unfortunately, the market peppers chaos in the mix and you have break outs and false break outs and market congestion. While the pits were alive and doing well, the locals

could use simple math to figure out price levels where the markets could potentially turn and reverse course or if these levels were penetrated with a certain level of gusto, then a momentum trade was setting up. These levels were labeled individual support and resistance levels (like the ones in the chart above). A resistance level is a price where price should struggle to cross above, and a support level is one where price should struggle to cross below. Combining the momentum and counter trend mechanisms into one strategy has been around since the 1980s. Richard Saldenberg along with John Ehlers/Mike Barna created very successful day trading systems with names like R-Breaker, R-Levels and R-Mesa. Many other traders utilized the same concepts and also created very successful algorithms – some leaned more on the either the breakout or counter trend entries. Initially, the success of these algorithms lay with the crowd mentality as many day traders in the pits and in front of the screens were looking at the same thing simultaneously – it was a self-fulfilling prophecy. However, as more traders started using these levels, the magic wore off. These levels became traps and the robustness of these day trading approach waned. Still the synergy of the breakout and counter trend approach was still a great approach and the only game in town. Traders started deriving their own levels and programmed entries and exits based on the interaction of the markets with these levels. Many traders incorporated risk measures by monitoring daily volatility and derived potentially higher probable trades by overlaying daily bar patterns.

The code that is used to develop these types of algorithms can be quite complicated, especially for a not-so experienced EasyLanguage programmer. For this reason, I have created this framework. The code is fully explained at the end of the manual and a video is included on using the framework to reflect your own observations on the chart and in code. The framework is not only a tool for developing day trade systems, but the logic and coding can take you a long way to becoming a much better EasyLanguage programmer.

## Contents

Introduction.....	1
Define Levels via Inputs .....	4
Support and Resistance Level Calculations.....	4
Midpoint Derived Inputs .....	4
Camarilla Inputs.....	5

Camarilla Zones explanation.....	5
Zone Analysis Framework Nomenclature .....	6
Crossing from Zone#1 to Zone#2 from Above Example .....	6
Keeping Track of Last <i>N</i> Level Crossings or Penetrations.....	6
In Which Zone did the Market Open [dayOpenZone] .....	6
Sequence Report .....	7
Using Sequence Report as a Tool for Further Research .....	7
How to Enter a Trade .....	8
Zone1 Break Out.....	8
Zone3 Counter Trend.....	8
Zone3 Hesitation.....	9
Zone 5 Break Out.....	9
Disclaimers.....	10
Results for past 5 years on the 4 entries.....	11
List of Names (Tokens) Used to Determine Path of Intraday Market Movement .....	13
Zone1 .....	13
Zone2 .....	13
Zone3 .....	13
Zone4 .....	13
List of Names (Tokens) Used to Determine Location of Daily Open Tick .....	14
How to Limit Duplicate Zone Entries.....	14
ZoneArray enumerations .....	14
Exits .....	15
SetStopLoss or SetProfitTarget.....	15
Connecting Entries with Exits.....	15
Code Explanation .....	16
Levels as Inputs Version .....	16
Other Inputs Used by the Framework.....	17
Variables Used in Framework .....	17
Trade Filters [1-3].....	18
Calculations Set on the First Bar of the Day.....	19
Crosses Below String Instantiations and Modifiers.....	21
Crosses Below String Instantiations and Modifiers.....	22
Extracting Last <i>N</i> Crossings .....	23
Trade Directive Nomenclature.....	24
Connecting Exits with Entries by Using Entry Signal Names.....	24

Built-In Profit Objective and Stop Loss .....	25
Report Generation.....	25
Some Test Examples of the ES.D for the past 5 years. ....	28
MidPoint Based Zones .....	28
Template #1 .....	28
Template #2.....	29
Unintended Consequences .....	32
Camarilla Based Zones.....	32
Package Contents .....	35

## ***Define Levels via Inputs***

The framework includes two templates:

- 6 zones derived by using the prior day's midpoint
- 8 zones derived by using the Camarilla equations.

Up to eight zones can be programmed by simply inputting the zone calculations directly into the Inputs of the strategy and indicator (you can add more zones by following the coding pattern).

Here are examples of the **Support** and **Resistance** levels derived by using yesterday's **midpoint** and the **Camarilla Zones** as inputs into an indicator and also a strategy:

## ***Support and Resistance Level Calculations***

### **Midpoint Derived Inputs**

```

Zone1TopForm(highD(1) + (highD(1) - (highD(1)+lowD(1))/2)),
Zone1BotForm(highD(1) + minMove/PriceScale),
Zone2TopForm(highD(1)),
Zone2BotForm((highD(1) + lowD(1))/2 + minMove/PriceScale),
Zone3TopForm((highD(1) + lowD(1))/2),
Zone3BotForm(lowD(1)),
Zone4TopForm(lowD(1) - minMove/PriceScale),
Zone4BotForm(lowD(1) - ((highD(1)+lowD(1))/2-lowD(1)));

```

## Camarilla Inputs

```
Zone1TopForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1 / 2),  
Zone1BotForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1/4 + minMove/priceScale),  
Zone2TopForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1/4),  
Zone2BotForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1/6 + minMove/priceScale),  
Zone3TopForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1/6),  
Zone3BotForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1/12 + minMove/priceScale),  
Zone4TopForm(CloseD(1)+(HighD(1)-LowD(1)) * 1.1/12),  
Zone4BotForm(CloseD(1)-(HighD(1)-LowD(1)) * 1.1/12 + minMove/priceScale),  
Zone5TopForm(CloseD(1)-(HighD(1)-LowD(1)) * 1.1/6),  
Zone5BotForm(CloseD(1)-(HighD(1)-LowD(1)) * 1.1/4 + minMove/priceScale),  
Zone6TopForm(CloseD(1)-(HighD(1)-LowD(1)) * 1.1/4),  
Zone6BotForm(CloseD(1)-(HighD(1)-LowD(1)) * 1.1/2 + minMove/priceScale);
```

## Camarilla Zones explanation

It may look like only seven zones are defined but **Zone0** and **Zone8** are the areas above **Zone1** and below **Zone7** respectively.

Each zone has an upper and lower boundary.

**Zone1Top** is defined as:  $\text{CloseD}(1) + (\text{Range}[1]) * 1.1 / 2$

**Zone1Bot** is defined as:  $\text{CloseD}(1) + (\text{Range}[1]) * 1.1/4 + \text{minMove}/\text{priceScale}$

**Zone2Top** is defined one tick below **Zone1Bot**

