

Easy Language



מה זה EasyLanguage ?

- **הינה שפת תכנות פשוטה, אך רבת**
עוצמה המשמשת לייצירת אינדיקטורים טכניים וסטרטגיות
מסחר בפלטפורמת מסחר - TradeStation
- **EasyLanguage מופעלת על ידי TradeStation**
- **כל אסטרטגיה וכל אינדיקטור המובנים ב TradeStation**
 כתובים ב **EasyLanguage** ונגישים לצפייה/שינוי.

מה זה ? EasyLanguage

סוגי אינדיקטורים וכלי ניתוח טכניים שנייתן ליצור ב
: TradeStation

- ▶ **Indicators**
- ▶ **ShowMe Studies**
- ▶ **PaintBar Studies**
- ▶ ActivityBar Studies
- ▶ ProbabilityMap Studies
- ▶ **Trading Strategies**
- ▶ **Functions**
- ▶ OptionStation Pricing Models
- ▶ OptionStation Search Strategies

שימוש במונחים מעולם המסחר

רעיוןנות מסחר לשפט מחשב בצורה פשוטה, ובשימוש טרמינולוגיה המשמשת את עולם המסחר/השוקות.

מידע הנכלי בכל בר/נر יפני :

- | | | |
|---------|----------|------------------------|
| ▪ Open | ▪ Date | ▪ Open Interest |
| ▪ High | ▪ Time | ▪ Options Related Data |
| ▪ Low | ▪ Volume | ▪ Fundamental Data |
| ▪ Close | ▪ Ticks | |

תהליך החישוב ב EasyLanguage

- תהליך החישוב ב EasyLanguage תמיד מתבצע בברים (או Ticks), ומתרחש בסגירה של הבר.
- תהליך החישוב מתבצע על התרשים/גרף משמאלי לימיין, החל מהבר השמאלי - הראשון עד הבר האחרון - הנוכחי.
- עם סגירת הבר כל התוכנית של האידיקטור/אסטרטגיה מופעלת במלואה.

תהליך החישוב ב EasyLanguage

MaxBarsBack □



EasyLanguage Punctuation

EasyLanguage is NOT case sensitive. Line returns, line spacing and paragraph indents are for readability only; they do not affect the EasyLanguage instructions.

- () Parentheses — used in statement structures; also for grouping mathematical operations.
- “ ” Quotes — denote a text item (e.g., Plot Name) within EasyLanguage.
- ; Semi-colon — indicates the end of a statement.
- :
- Colon — denotes the declaration of a list.
- ,
- Comma — separates items in a list.
- [] Square brackets — used to reference data from a previous bar and to displace a plot; also used in arrays.
- { }
- Curly brackets — any text between the brackets is notation (remarks) and not part of the EasyLanguage instructions.
- // Double slash — any text following the double slash, for the remainder of that line only, is notation (remarks) and not part of the EasyLanguage instructions.

אינדיקטור - Indicator

- אינדיקטור הינו טכנית ניתוח שמשתמשת במידע/נתוני מסחר ובאמצעות חישוב מתמטי מציגה את התוצאה על גבי התרשים/גרף המציגים
- אינדיקטור יכול להיות מוצג כמעט בכל הכלים של RadarScreen, Chart-TradeStation, Scanner ו OptionStation

Plot Statement

PlotN(numeric expression, “plot name”);

N = 1 to 99

Numeric Expression is the value to be plotted; stated another way, it is the location of the plot on the y-axis for each bar (x-axis).

Plot name is the label that will be used for this plot in the formatting dialogs and the Data Window.

Note: including “plot name” in a plot statement is optional, but recommended. Other optional parameters for formatting are also available.

Ex-01 Close

Write an Indicator plots a line connecting the closing prices of the bars in a chart

Ex-02 Volume

Write an Indicator that plots the Volume
(shares)

Ex-02 Volume

Write an Indicator that plots the Volume
(shares or contracts) or the Tick Count
(transactions), depending on the type of chart
and the formatting settings

Ex-03 High and Low

Write an Indicator that plots a line connecting the highs of each bar and a second line connecting the lows of each bar.

Set the properties to:

display the plots as lines, and display the plots in the same sub-graph as the price data.

Mathematical Operators

Addition +

Subtraction -

Multiplication *

Division /

Ex-04 Real Body

Write an Indicator that plots the difference – “Real Body”, between the Close and the Open of each bar.

This will be a positive value if the close is greater than the open and a negative value if the close is less than the open.

Order of Operations

TradeStation calculates arithmetic operations as follows:

Multiplication and division, from left to right followed by **Addition** and subtraction, from left to right.

This order may be customized by use of **parentheses**.

$$1 + 5 * 3 + 4 = 20$$

$$(1 + 5) * (3 + 4) = 42$$

$$1 + (5 * (3 + 4)) = 36$$

Ex-05 Mid Price

Write an Indicator that plots a line connecting the middle price of each bar.

Ex-06 Volume Weighted Range

Write an Indicator that calculates the range of each bar multiplied by its volume, resulting in a volume-weighted range indicator

Easy Language



Referencing Data from Previous Bars

ערכים מחיר קודמים/היסטוריה וערכים אחרים ניתנים לגישה באמצעות אינדקס בסוגרים מרובעים [ח] המוצמדים לשם המשתנה או הפעטטר (מילה השמורה) של הערך להפניה.

The close of the bar 5 bars ago:

Close[5]

Range of the bar 12 bars ago:

High[12] - Low[12] or **(High - Low)[12]**

but not High - Low[12] !!

Ex-07 Net Change Oscillator

Write an Indicator that plots a 5-bar Net Change (close to close) Oscillator.

Ex-08 Bands

Write an Indicator that plots two bands around the price data.

- ❑ One band adds half the previous bar's range to the current bar's opening.
 - ❑ The other band subtracts half the previous bar's range from the current bar's opening.
-
- Remember: The range of a bar is High – Low.

This Indicator should overlay the price bars

משתנים - Variables

- ▶ משתנים הינם כלי תכנתית המאפשרים לשמר ערכי נתונים המתקבלים במהלך החישוב ולהתייחס לערכים אלה בעת הצורך
- ▶ לכל יש משתנה שם ייחודי
- ▶ משתנים יכולים להיות משתנים מספריים, משתנים לוגיים (True/False), או משתנה מחרוזת - טקסט .
- ▶ שימוש במשתנים מאפשר לארגן את הקוד על ידי מתן למשתנים שמות שימושיים המתארים את אופי החישוב או המטרה של נתונים/ערכים מאוחסנים.

משתנים - Variables

Variables:

```
LastHigh(0),      { creates a numeric variable }  
NewHigh(false),   { creates a true/false variable }  
HighAlert("");    { creates a text variable }
```

השמה למשתנים - Assignment

Vars:

```
SlowAverage(0), FastAverage(0), MyParam (0);
```

```
MyParam = 2;
```

```
MyParam = (SlowAverage + FastAverage)/2;
```

```
SlowAverage = Average(Close, 18);
```

```
FastAverage = Average(Close, 9);
```

סוגי משתנים - Variables Types

There are three variable, input, and array types:

- ❑ **Numeric** – Holds a simple or complex number, positive or negative. There are three numeric types in EasyLanguage: **Integer**, **Float**, and **Double**.
- ❑ **True/false** – Holds a true/false state, either a true/false expression or the words true or false.
- ❑ **Text String** – Holds a text string, numbers or letters enclosed in quotation marks (“text”).

Pre Declared Variables

- ▶ Pre-declared variables are variables that do not need to be declared and automatically recognized by the language.
- ▶ Pre-declared variables come in two types: numeric and true/false.
- ▶ **Value0** to **Value99** – There are 100 pre-declared numeric variables.
`Value1 = Average(Price, Length);`
- ▶ **Condition0** to **Condition99** – There are 100 pre-declared true/false variables.
`Condition1 = Value1 >= MyAverage;`

Historical Reference of Variables

- ▶ משתנים, פונקציות, ומילוט שמות ניתן להתייחס לערבי ברים קודמים/היסטוריים של עצם בגרף, באמצעות הוספת הסוגר מרובע [ח] לאחרי שם המשתנה
- ▶ ניתן לפנות אליו אופן, על ידי שימוש ב[ח], לערכים ההיסטוריים באלמנטי מערך.

דוגמאות:

- ▶ `Plot1(Close[5]);`
- ▶ `Plot2((Close – Open)[1]);`
- ▶ `Condition1 = Condition2[1] OR Condition2[2];`

Ex-09 Bands2

This is a re-write *08 Bands, incorporating user-declared numeric variables.

Write an Indicator that plots two bands around the price data.

- One band adds half the previous bar's range to the current bar's opening.
- The other band subtracts half the previous bar's range from the current bar's opening.

Ex-10 Net Change Oscilator-2

This is a re-write of *07 NetChangeOsc, incorporating user-declared numeric variables

Write an Indicator that plots a 5-bar Net Change Oscillator.

לולאת FOR

```
for idx=<init_value> to <final_value> begin  
    .  
    . { loop body...}  
    .  
end;
```

- ▶ A for loop repeats one or more statements a specific number of iterations defined by the user in the for statement loop range values.
- ▶ This numeric range is incremented in a loop counter stored in a variable for reference within the block of statements.

לולאת FOR

- ▶ Once the counter reaches the specified limit, the for loop ends and the next statement in the code is evaluated.
- ▶ For loops can iterate the count either ascending or descending depending on which reserved word is used:
 - To – count ascending
 - DownTo – count descending

לולאות FOR

Example (ascending):

```
Vars: RangeSum(0), x(0);{ Declare Variables }
RangeSum = 0; { Reset variable each bar}
```

```
For x = 0 to 5 begin
    RangeSum = RangeSum + Range[x];
end;
```

This for loop will iterate through the statement block 6 times (0, 1, 2, 3, 4, 5).

לולאות FOR

Example (descending):

```
Vars: RangeSum(0), x(0);{Declare Variables}
RangeSum = 0; {Reset variable each bar}
```

```
For x = 5 downto 0 begin
    RangeSum = RangeSum + Range[x];
end;
```

This for loop will iterate through the statement block 6 times (5, 4, 3, 2, 1, 0).

שימוש בפונקציות – Using Functions

פונקציות הן מבנה תוכנה הממשות חישוב/נוסחה והמחזירות ערך מחושב.

הפונקציות נראות לשימוש בטכנית ניתוח או באסטרטגיה באמצעות מילוט מפתח – שם הפונקציה ו לבטל את הצורך לזכור כל פעם מחדש נוסחאות מורכבות

הפונקציות הנפוצות ביותר בשפה הן חישוב ממוצע, סטיית תקן, RSI, ADX

פונקציות רבות דורשות מהמשתמש לספק פרמטרים לפונקציה

שימוש בפונקציות – Using Functions

- ▶ Functions are frequently used formulas (or comparisons) that return numeric values (or true or false). They may be called for use in any Analysis Technique or Strategy with just a few words,
- ▶ eliminating the need to re-create complex formulas.
- ▶ • Functions can be used in any set of EasyLanguage rules or instructions.
- Most common mathematical and statistical formulas are stored as Functions, such as

Ex-11 Momentum

Write an Indicator that plots the 10-bar momentum of closing prices.

Ex-12 Real Body Average

Write an Indicator that plots the 10-bar average of the difference between the Close and the Open of each bar, as originally plotted in *04 Real Body.

Ex-13 Envelope

Write an Indicator that plots a 20-bar moving average of the highs and a 20-bar moving average of the lows.

Select names and declare variables for the two moving average values.

Hint: The "Average" function may be dragged in from the Dictionary.

פרמטרי קלט – Inputs

- הצחרת פרמטרי `utu` בטכנית ניתוח או אסטרטגיה נוספת פרמטרי קלט שבאמצעותם ניתן להזין ערכי קלט משתנים, ומאפשר בכר לשנות את אופי/נתוני החישוב החישוב מוביל לשנות את קוד ה- `EasyLanguage`
- שימוש בפרמטרי `utu` מאפשר ליצור פונקציות גנריות, ומאפשר לבחון בקלות ביצוע החישוב על סenarios שונים מוביל לשנות את קוד `EasyLanguage`
- פרמטרי `utu` משמשים להזנת ערכי קלט לחישוב, בקרה וקבעת אופן ביצוע החישוב, קביעת אופן הצגה ומיפוי הצבעים, אופטימיזציה של אסטרטגיית ועוד ...

פרמטרי קלט – Inputs

הצורה על פרמטרי Input :

Input: InputName(default value);

- ❑ לכל Parameter Input ניתן ערך ברירת מחדל (מספרי, text, True/False), המגדיר את טיפוס המשתנה.
- ❑ לא ניתן לשנות את ערכי ה Input מתוך הקוד של טכניקת Format Analysis או אסטרטגיה, אלא מתוך חלון Techniques and Strategies

❑ דוגמא:

Input: Factor(1.005);

Input: Price(Close);

Ex-14 Envelope2

Learning objective: Using Inputs in place of fixed values.

Re-write Indicator 13 incorporating Inputs for the lengths of the moving averages.

Ex-15 Trailing Hi Lo

Learning objective: Using Inputs; using the functions Highest and Lowest.

Write an Indicator that plots the highest value of an Input and the lowest value of an Input over the last n bars.

Relational Operators

True/False משמשים לשערוך ביטוי ישותם שמונה בEasyLanguage :Related Operators

- < Less than
- <= Less than or equal to
- > Greater than
- >= Greater than or equal to
- = Equal to
- <> Not equal to

Easy Language



Relational Operators

❑ משמשים לשערוך ביטוי True/False
❑ ב EasyLanguage ישנו שמהןoperators :Related

- < Less than
- <= Less than or equal to
- > Greater than
- >= Greater than or equal to
- = Equal to
- <> Not equal to

Relational Operators

בנוסף לשישה יש ב EasyLanguage שני אופרטורים חודים שמשמשים להשוואות על ברים עוקבים :

Crosses over (same as crosses above)

returns True if

$x > y$ on current bar and $x[1] < y[1]$ previously

Crosses under (same as crosses below)

returns True if

$x < y$ on current bar and $x[1] > y[1]$ previously

Logical Operators

אופרטורים לוגיים מאפשרים לשלב מספר ביטויים בוליאניים של True/False לכדי ביטויי מורכב לוגי מורכב

AND

both expressions must be true for the entire expression to be true

$High > High[1] \text{ AND } Close > Close[1]$

OR

if either expression is true then the entire expression is true

$High > High[1] \text{ OR } Close > Close[1]$

ביצוע מותנה - Conditional Branching

If Then

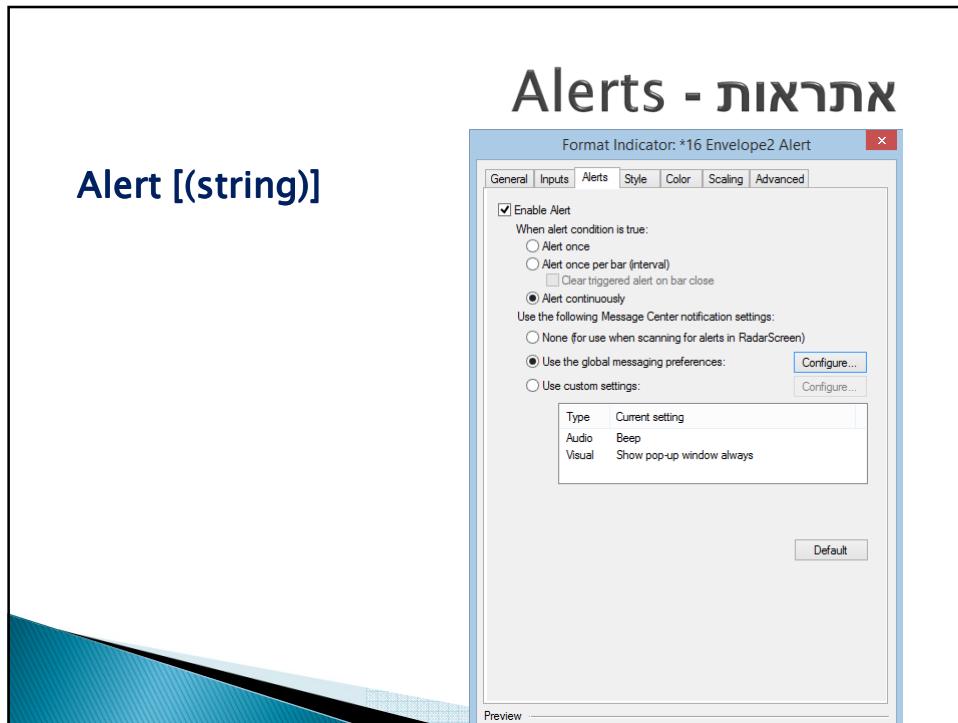
```
if Condition then  
<Your Code Line1>
```

```
if Condition then begin  
<Your Code Line1>  
<Your Code Line2>  
end
```

ביצוע מותנה - Conditional Branching

If Then Else

```
if Condition then  
<Your Code Line1>  
else  
<Your Code Line2>;
```



Ex-16 Envelope2 Alert

Learning objective: Using If/then statements to set alert criteria

Re-write Indicator 14 – Envelope2 incorporating conditions for alerts

Ex-16 Envelope2 Alert

```
Input: UpperLength(20), LowerLength(20);  
Vars: UpperLine(0), LowerLine(0);  
UpperLine = Average(High, UpperLength);  
LowerLine = Average(Low, LowerLength);  
Plot1(UpperLine, "UpperLine");  
Plot2(LowerLine, "LowerLine");  
  
//Alert command is placed inside an If/then statement  
If Low > UpperLine then  
    Alert;  
  
//Alert command is placed inside an If/then statement  
If High < LowerLine then  
    Alert;
```

Ex-17 Mov Avg & Bands

Learning objective: Using If/then statements to set alert criteria; review Inputs, Variables and Functions.

Write an Indicator that plots 3 lines:

- 1) a moving average
- 2) the same moving average plus 1 standard deviation in price
- 3) the same moving average minus 1 standard deviation in price.

Alerts are set for breaks of either of the outer bands, and custom Alert strings are included.

Ex-17 Mov Avg & Bands

```
Inputs: Price(Close), Length(18);

Vars: MAvg(0), StDv(0), HiBand(0), LoBand(0);

MAvg = Average(Price, Length);
StDv = StdDev(Price, Length);
HiBand = MAvg + StDv;
LoBand = MAvg - StDv;

Plot1(MAvg, "Avg");
Plot2(HiBand, "HighBand");
Plot3(LoBand, "LowBand");

//Alert command is in an If/then statement
//Alert message follows in parentheses
If Close crosses over HiBand
    then Alert ( "Close has crossed over HiBand " + Description );

//Alert command is in an If/then statement
//Alert message follows in parentheses
If Close crosses under LoBand
    then Alert ( "Close has crossed under LoBand " + Description );
```

PaintBar Studies

- טכניקת ניתוח המיעדרת לסימון/"צביית" גוף הבר (או חלקו) על גրף המחירירים.
- משמשת ליזיהו/ איתור אירועים ותבניות וסימון הבר המיכון לאירוע זה בגרף המחירירים
- פקודת הדפסה (**plot statement**) מינוחית משמשת לסימון ב :PaintBar

PlotPB(*StartPaint*, *EndPaint*, [*"plotname"*]);

NoPlot(*n*);

PaintBar Studies



Ex-20 MomentumPositive

Learning objective: Writing a PaintBar study using the PlotPB statement

Write a PaintBar that marks those bars on which the Momentum is greater than 0.

Ex-20 MomentumPositive

```
Input: Price(Close), Length(10);

Vars: Mom(0);

Mom = Momentum(Price, Length);

If Mom > 0 then begin
    PlotPB(High, Low, "MomPos");
    Alert;
end
```

Bar Date

- כל בר בגרף נושא תווית של תאריך – Bar Date לפני הפורט של EL.

Date Format: 1|Year|Month|Date

- 2 ספרות של השנה - Year
- 2 ספרות של החודש - Month
- 2 ספרות של התאריך - Date

דוגמא: התאריך 27/07/2015 יסומן 1150727

Bar Date

- המילה השמורה Date מוחזירה את התאריך של הבר

- בר בגרף Weekly יסתיים תמיד ב FRI , ובר גרפ Monthly יסתיים תמיד ביום המසחר האחרון באותו חודש.

דוגמא: תנאי לגילוי שינוי תאריך בבר הנוכחי:
if (Date <> Date[1]) then

Bar Time

כל בר בגרף נושא תווית של זמן סיום – Bar Time – לפי הפורט של EL.

Time Format: Hours|Minutes (ללא נקודותים)
Minutes 2 ספנות ל Hours ו 2 ספנות ל Minutes

דוגמא: השעה 15:10 תסומן 1510

Referencing a Bar

כל בר ב- EL ניתן ליחס על ידי שלושה פרמטרים:

- Date
- Time
- BarNumber

- (n) BarNumber - פונקציה המחזירה את המספר הסידורי של הבר אחרי ערך ה `.MaxBarsBack`.
- n - אינדקס המתיחס לבר לפני הבר הנוכחי

Ex-21 My Day

Learning objective: Writing a PaintBar study referencing date in EasyLanguage.

Write a PaintBar that marks each bar that occurs on the day specified by the Input.

Ex-21 My Day

```
Input: DoW(Wednesday);  
  
If DayOfWeek(Date) = DoW then  
    PlotPB(High, Low, "DoW");  
//Reserved word "DayOfWeek" finds the day for a specific date
```

Easy Language



PaintBar Studies



Ex-20 MomentumPositive

Learning objective: Writing a PaintBar study using the PlotPB statement

Write a PaintBar that marks those bars on which the Momentum is greater than 0.

Ex-20 MomentumPositive

[Update Value Intra-Bar (Tick by Tick)]

```
Input: Price(Close), Length(10);

Vars: Mom(0);

Mom = Momentum(Price, Length);

If Mom > 0 then begin
    PlotPB(High, Low, "MomPos");
    Alert;
end
else
    NoPlot(1);
//NoPlot unpaints the bar if the condition changes during the bar
```

Creating Trading Strategies

- אסטרטגיית מסחר הינה מערכת של כללי מסחר המבוצעת כניסה ויציאה מתוכנתים לפוזיציות מסחר.
- BackTesting מספקת כלים ומנגנון להערכת TradeStation היסטורי של האסטרטגיה, המאפשרת לבדוק ולהעריך את ביצועים ההיסטוריים, לבצע אופטימיזציה, ואוטומציה בביצוע פקודות בשלב מסחר אמיתי עם האסטרטגיה
- אסטרטגיה מכילה פעולות מסחר אחת לפחות. גרף - Chart יכול להכיל ולבצע מספר אסטרטזיות בו בזמןית.

Creating Trading Strategies

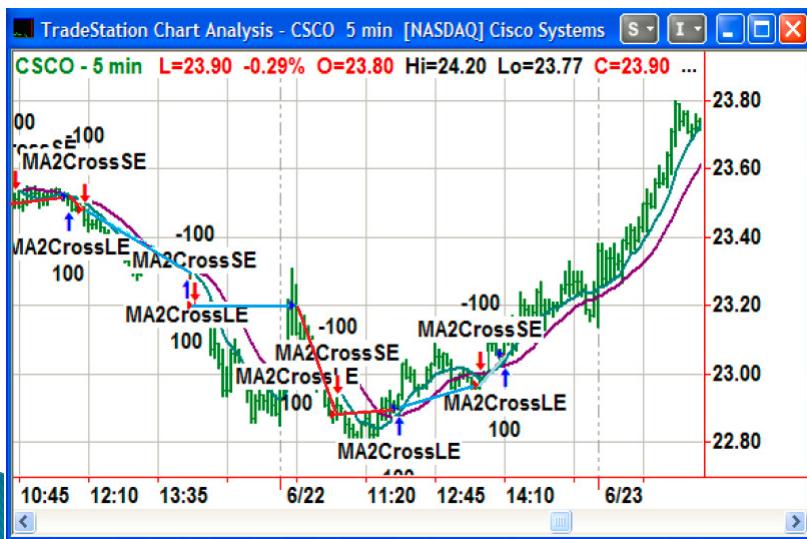
- Buy [("Order Name")] [Number of Shares/Contracts] [Order Action];
- SellShort [("Order Name")] [Number of Shares] Execution Method ;

דוגמא:

```
if Average(Close,9) Crosses above the Average(Close,18) then
    Buy ("MA2CrossLE") next bar 100 shares at Market;

if Average(Close,9) Crosses Below the Average(Close,18) then
    Sell Short("MA2CrossSE") next bar 100 shares at Market;
```

Creating Trading Strategies



Creating Trading Strategies

Strategy Engine Calculation

On Bar Close - האסטרטגיה מחושבת רק פעם אחת עבור כל

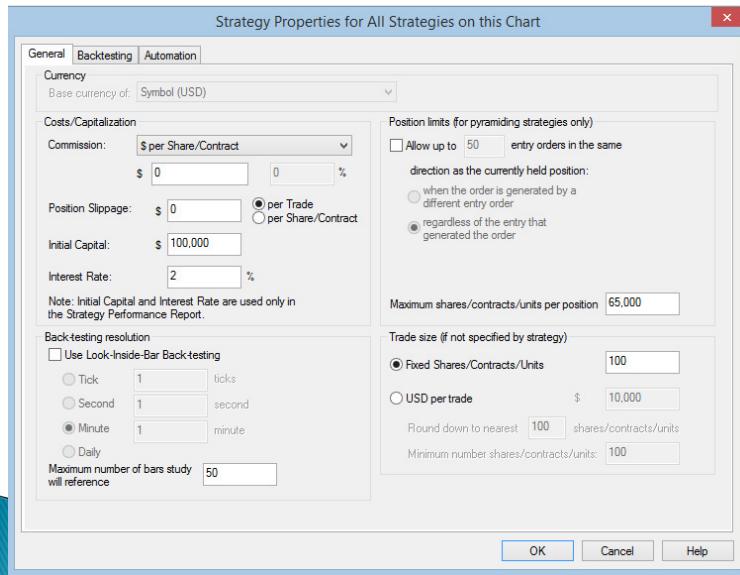
בר, בסגירה של הבר. במידה ומתקיימים התנאים פקודות מסחר מיצירות עבור הבר הבא, וمبוטלות במידה ולא הופעלו במשך הבר הבא.

Intra Bar - מבחינה הסטורית, האסטרטגיה מחושבת פי

Look Bar Back-Testing
Inside

בהרצת האסטרטגיה בזמן האמת, האסטרטגיה מחושבת על בסיס Tick. במידה ומתקיימים התנאים פקודות מסחר מיצירות עבור הבר הבא, וمبוטלות במידה ולא הופעלו במשך הבר הבא.

Strategy Properties



Strategy Properties

Costs/Capitalization

- Commissions & Slippage
- Initial Capital
- Interest Rate

Back-testing Resolution

- **Look-Inside-Bar Back-Testing**
- **Max Number of bars study will reference**

Strategy Properties

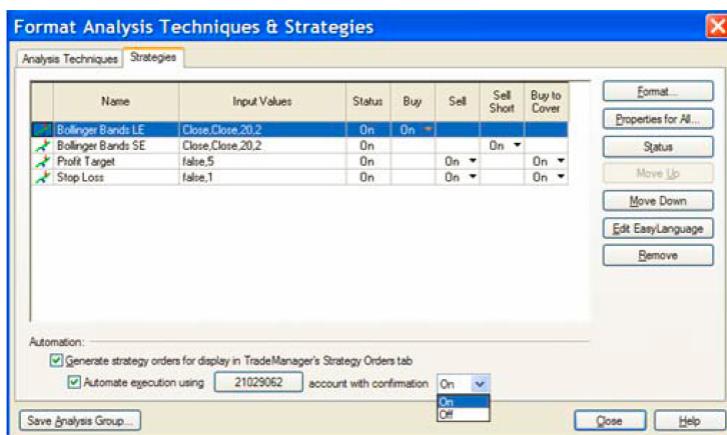
Position Limits

- Num of Entry orders in the same direction
- Position Size – Max Shares/Contracts/Units

Trade Size

- Fixed Shares/Contracts/Units vs \$ Per Trade

Strategy Automation



Strategy Order Syntax

- ❑ **Buy:** Establish or add to a long position. Any existing short position will be covered entirely before the long position is established. Two orders are generated.
- ❑ **SellShort or Sell Short:** Establish or add to a short position. Any existing long position will be liquidated entirely before the short position is established. Two orders are generated.

Example: Our Trend Following Strategy

Strategy Order Syntax

- ❑ **Sell:** Liquidates a long position only. Can never establishes a short position.
- ❑ **BuyToCover or Buy To Cover:** Cover a short position only.

Strategy Order Syntax

Order Actions

- ❑ **next bar at Market:** Market order at the open of the next bar or the next tick.
- ❑ **next bar Stop:** Market order on the next bar when the stop price is reached. All stop orders in Easy-Language are Stop Market.
- ❑ **next bar Limit:** Limit order on the next bar if the limit price is reached.

Strategy Order Syntax

Order Actions

- ❑ **this bar on Close:** Market order on the close of this bar, generally used for historical backtesting purposes only. This will not generate a market on close order.

Usage Examples:

- Buy next bar at Market; Buy (“**Le1**”) next bar at Market;
- Sell Short next bar 50 Limit; Sell Short (“**Sh1**”) next bar 50 Limit;
- Sell next bar 50 Stop;
- Buy to Cover this bar on Close;

Orders – Signal Names

Usage Example:

Buy ("L1LE") this bar on Close;
SellShort("L1SE") next bar at Market;

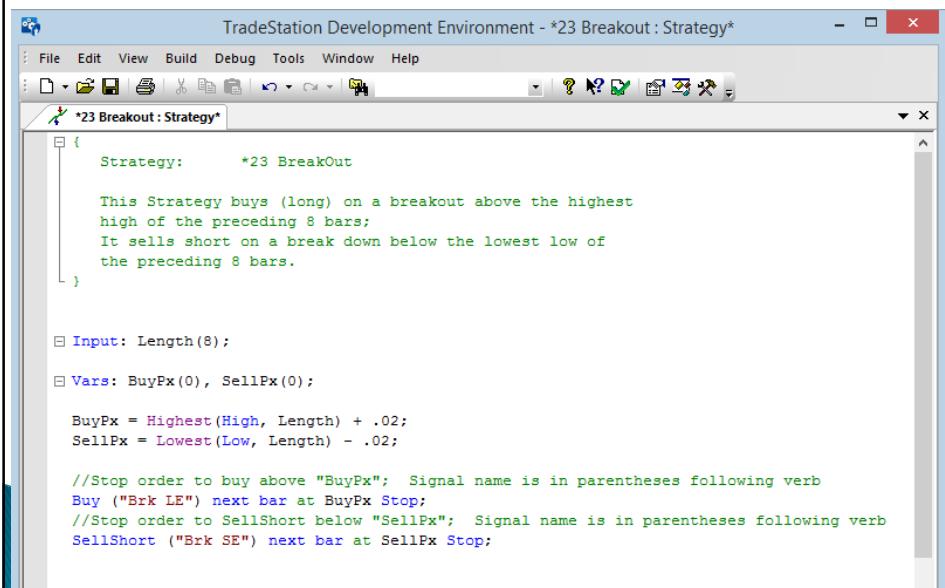


Ex-23 Breakout

Learning objective: Writing a Strategy; stop order syntax.

Description: Write a Strategy that buys (long) on a breakout above the highest high of the preceding 8 bars; it sells short on a break down below the lowest low of the preceding 8 bars.

Ex-23 Breakout



```

TradeStation Development Environment - *23 Breakout : Strategy*
File Edit View Build Debug Tools Window Help
*23 Breakout : Strategy*
{
    Strategy:      *23 BreakOut

    This Strategy buys (long) on a breakout above the highest
    high of the preceding 8 bars;
    It sells short on a break down below the lowest low of
    the preceding 8 bars.
}

Input: Length(8);

Vars: BuyPx(0), SellPx(0);

BuyPx = Highest(High, Length) + .02;
SellPx = Lowest(Low, Length) - .02;

//Stop order to buy above "BuyPx"; Signal name is in parentheses following verb
Buy ("Brk LE") next bar at BuyPx Stop;
//Stop order to SellShort below "SellPx"; Signal name is in parentheses following verb
SellShort ("Brk SE") next bar at SellPx Stop;

```

Ex-23 Breakout



Strategy Info – Market Position

`MarketPosition(N)` returns whether the strategy is currently flat, short, or long on the current bar or for N closed positions ago. `MarketPosition` return values are:

- -1 for a short position.
- 1 for a long position.
- 0 for flat (no position).

Usage Example:

```
if MarketPosition = 0 then  
    Buy next bar at Market;
```

Strategy Info – Market Position

Historical Reference of Strategy Position Reserved Words

Usage Example:

```
Vars: MP(0);  
MP = MarketPosition;  
if MP[2] = -1 AND MP[1] = 0 then  
    Buy next bar at Market;
```

Strategy Info – Entry Price

EntryPrice – returns the entry price for the current position.

It can also report what the entry price was N closed positions ago – **EntryPrice(N)**.

Usage Example:

```
if MarketPosition = 1 then  
    Sell next bar at EntryPrice - .10 Stop;
```

Strategy Info – Bars Since Entry

BarsSinceEntry – returns the number of bars from the entry bar for the current position.

It can also report how many bars from the entry bar of N closed positions ago –

BarsSinceEntry(N)

Strategy Info – Bars Since Entry

BarsSinceEntry returns the number of bars from the entry bar for the current position.

Usage Example 1:

```
if MarketPosition = 1 AND BarsSinceEntry > 5 then  
    Sell next bar at Market;
```

Usage Example 2:

```
if MarketPosition = 1 AND BarsSinceEntry > 5 then  
    Sell next bar at Low[BarsSinceEntry] Stop;
```

Additional Strategy Position Info

- AvgEntryPrice
- BarsSinceExit
- Current Shares
- Current Contracts
- EntryDate
- EntryTime
- ExitDate
- ExitTime

Ex-24 Mov Avg Cross

Learning objective: Writing a Strategy; using Strategy Position Reserved Words.

Description: Write a Strategy that takes long and short positions based on the crossovers of two moving averages.

A position is closed when it has been held at least a minimum number of bars and is not making at least a minimum profit.

Ex-24 Mov Avg Cross

```
*24 Mov Avg Cross :Strategy*
{
    Strategy:      *24 Mov Avg Cross

    This Strategy takes long and short positions based on the crossovers of two moving averages.
    A position is closed when it has been held at least a minimum number of bars and is not
    making at least a minimum profit.
}

Input: ShortLen(9), LongLen(18), MinHold(8), MinProf(50);

Vars: ShortMA(0), LongMA(0);

ShortMA = Average(Close, ShortLen);
LongMA = Average(Close, LongLen);

If ShortMA crosses over LongMA then
    Buy next bar at market;
If ShortMA crosses under LongMA then
    SellShort next bar at market;

If BarsSinceEntry > MinHold and OpenPositionProfit < MinProf then begin
    Sell ("EL-M1") next bar at market;
    //Sell and BuyToCover are in a Block If/then since they are dependent
    BuyToCover ("ES-M2") next bar at market;
    //on the same conditions, and mutually exclusive
end;
```

Ex-24 Mov Avg Cross



Strategy Performance Info – NetProfit

NetProfit returns the cumulative net profit or loss for all closed trades in the chart; this is the closed trade equity curve value for each bar. The value will either be positive, negative, or zero.

Usage Example :

Vars: TradeSize(0);

TradeSize = 1000;

TradeSize = TradeSize + (**NetProfit** / Close);

Buy next bar TradeSize Shares at Market;

Strategy Performance Info – OpenPositionProfit

OpenPositionProfit returns net profit or loss for the current open position.

Adding **OpenPositionProfit** to **NetProfit**, derives the same value as the detailed equity curve on a bar-by-bar basis.

The value will either be positive, negative, or zero.

Usage Example:

Input: ProfitExit(200);

```
if OpenPositionProfit >= ProfitExit then begin  
    Sell next bar at Market;  
    BuyToCover next bar at Market;  
end;
```

Additional Strategy Performance Info

- GrossProfit
- GrossLoss
- NumWinTrades
- NumLosTrades
- PercentProfit
- Total Trades

Ex-25 Momentum Cross

Learning objective: Using user-declared true/false variables; MRO—Most Recent Occurrence Function

Description: Write a Strategy that:

- ❑ Buys (long) when Momentum crosses over 0, as long as it has not crossed under 0 within the last 4 bars; it sells (liquidates long positions) if Momentum declines on 2 consecutive bars;
- ❑ Sells short when Momentum crosses under 0, as long as it has not crossed over 0 within the last 4 bars; it buys to cover (covers short positions) if Momentum rises on 2 consecutive bars.

Ex-25 Momentum Cross

MRO—Most Recent Occurrence Function

MRO (Function)

 [Disclaimer](#)

The **MRO** (Most Recent Occurrence) function returns the number of bars ago the specified expression was **True**. Or, if the specified expression did not occur within the last **x** number of bars, the function informs you of such.

Syntax

`MRO (Test, Length, Instance)`

Returns (Integer)

A numeric value containing the number of bars ago that the specified Expression was **True**; -1 if Expression was not found to be **True** within the last **Length** bars.

Parameters

Name	Type	Description
Test	TrueFalse	Sets the true/false expression to check for (that is, Close > Open).
Length	Numeric	Sets the number of bars to check.
Instance	Numeric	Sets which occurrence, for example, 1 = most recent, 2 = 2nd most recent, and so on.

Ex-25 Momentum Cross

```
*25 Momentum Cross : Strategy*
Input: Length(10);

Vars: Mom(0), BullCx(false), BearCx(false);
//BullCx and BearCx are user-declared true/false Variables

Mom = Momentum(Close, Length);
BullCx = Mom crosses over 0;
//A true false expression must be assigned to a true/false Variable
BearCx = Mom crosses under 0;

If BullCx and MRO(BearCx,4,1) = -1 then
//MRO function checks that there was no BearCx in the last 4 bars
Buy next bar at Close of this bar limit;

If BearCx and MRO(BullCx,4,1) = -1 then
//MRO function checks that there was no BullCx in the last 4 bars
SellShort next bar at Close of this bar limit;

If Mom < Mom[1] and Mom[1] < Mom[2] then
//Exit long position after 2 consecutive bars of weaker momentum
Sell next bar at market;

If Mom > Mom[1] and Mom[1] > Mom[2] then
//Exit short position after 2 consecutive bars of stronger momentum
BuyToCover next bar at market;
```

Ex-25 Momentum Cross



Built-in Stops

- ❑ **SetDollarTrailing** – sets an exit stop a fixed number of dollars away from the peak profit.
- ❑ **SetPercentTrailing** – sets an exit stop a fixed percent of the peak profit away from the peak profit, after a minimum profit is achieved.

Built-in Stops

- ❑ **SetProfitTarget** – sets an exit order at a fixed dollar profit target.
- ❑ **SetStopLoss** – sets a stop loss order at a fixed dollar risk from entry.
- ❑ **SetBreakEven** – sets an exit stop at the entry price, after a minimum profit is achieved.

Built-in Stops

Usage Example:

```
Vars: MoValue(0);
```

```
MoValue = Momentum(Close, 10);
if MoValue crosses over 0 then
    Buy next bar at Market;
SetStopLoss(100);
SetProfitTarget(100);
```

Built-in Stops

SetStopPosition – exit is calculated for the entire position in dollars.

SetStopShare or **SetStopContract** – exits are calculated per share or contract.

Usage Example:

```
Inputs: StopAmt(1), ProfitAmt(1);
```

SetStopShare;

SetStopLoss(StopAmt);

SetProfitTarget(ProfitAmt);

הכנסות Trailing Stop



Ex-26 Key Reversal

Write a strategy that uses key reversals up and key reversals down to identify entry points.
Declare and assign Variables for key reversals up and down.

Have the strategy enter a limit order to buy on the bar following a key reversal up, at a limit price better than the current bar's close.

Ex-26 Key Reversal

Have the strategy enter a limit order to sell short on the bar following a key reversal down, at a limit price better than the current bar's close.

Declare an Input for the number of points above or below the reversal bar's close to set the limit order prices; have the Input default to 5 points.

Ex-26 Key Reversal

```

≡ Input: LimitPoints(.05);

≡ Vars: RevUp(false), RevDown(false);

RevUp = Low < Low[1] and Close > Close[1]; //True/false Variable
RevDown = High > High[1] and Close < Close[1];

If RevUp then
    Buy next bar at Close of this bar - LimitPoints limit; //Position
                                                //the bar after the reversal
If RevDown then
    SellShort next bar at Close of this bar + LimitPoints limit;

```

Commentary (Reserved Word)

This reserved word sends the expression (or list of expressions) to the Analysis Commentary window for whatever bar is selected on the price chart.

Example:

```
Commentary("This is one line of commentary") ;
Commentary("The 10-bar avg = ", Average(Close, 10), Newline);

Commentary("Fast_Avg=", Fast_Avg, "Slow_Avg=", Slow_Avg),
Newline);
```

RSI - מדד העוצמה היחסית

- **Relative Strength Index - RSI** . פותח בשנת 1978 על ידי וילדר.
- הממדד אומד את העוצמה בין ממוצע הרווח לבין ממוצע ההפסד במדד נייר השער על פני מספר תקופות נתון (וילדר המליץ על תקופה של 14)
- מדד זה RSI נוע על סקלת הערכאים בין 0-[שיא החולשה] ל 100-[שיא העוצמה] כאשר קו 50 הינו קו האמצע.

$$\text{RSI} = 100 - \frac{100}{1 + RS}$$

RS = Average Gain / Average Loss

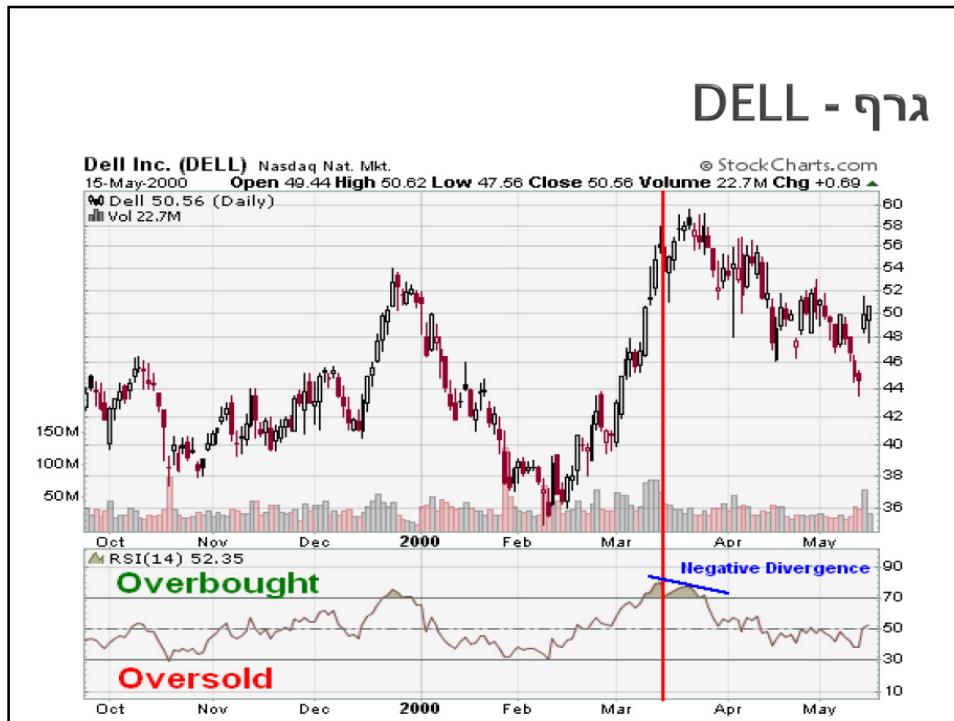
- Average Gain = [(previous Average Gain) x 13 + current Gain] / 14.
- Average Loss = [(previous Average Loss) x 13 + current Loss] / 14.

RSI - מדד העוצמה היחסית - שימושים

- RSI- 70 ומעלה מכונה רמת קנית יתר (overbought) ומצביע על התחזקות המניה.
- RSI - 30 ומתחת המכונה רמת מכירת יתר (oversold) ומצביע על החלשות המניה.
- עלייה במדד מעל 30 נחשבת לאיות שורי, וירידה מתחת ל 70 נחשבת לאיות דובי.
- עלייה מעל 50 בערך המדד נחשבת לאישוש לנטייה שורית (העוצמת הרווחים הממוצעת גדולה מעוצמת ההפסדים הממוצעת) וירידה מתחת ל 50 לאישוש לנטייה דובית.

RSI - מדד העוצמה היחסית - שימושים

- התבדרות (divergence) בערכי המדד בין ערכי נייר הערך באזורי קנית/מכירת יתר מהוות איות אמין יותר להיפוך מגמה .





EMC - מדד העוצמה היחסית - RSI



DNA - מדד העוצמה היחסית - RSI



Ex-28 RSI Overobught Oversold

- ▶ Learning objective: Using Built-in Stops in custom Strategies; using HighestBar and LowestBar Functions.
- ▶ Description: This strategy takes long (short) positions when the oversold (overbought) RSI begins to turn. For longs, RSI must be below 50 and a 7-bar low must have been made at least 3 bars ago.
- ▶ Shorts are the reverse. Built-in Stops have been added.

Ex-28 RSI Overobught Oversold

```
□ Input: Length(14), StopAmt(50), BEAmt(50), TrlgAmt(100);  
□ Var: xRSI(0);  
  
xRSI = RSI(Close, Length);  
  
If xRSI < 50 and LowestBar(xRSI, 7) >= 3 then  
    Buy next bar at market;  
  
If xRSI > 50 and HighestBar(xRSI, 7) >= 3 then  
    SellShort next bar at market;  
  
SetStopLoss(StopAmt);  
SetBreakEven(BEAmt);  
SetDollarTrailing(TrlgAmt);
```