

Instructions	
Product:	!QCL.MyIndex (extended version)
Date:	March 28, 2013
Purpose:	Create a custom index like the Dow Jones Industrial Average, NASDAQ-100 or S&P-100. It simplifies custom index creation with symbols, multipliers and base values specified in comma-separated inputs. Index data can be written to file and used as 3 rd -party data.
Contact Information:	!QCL Solutions
Document Notes:	<i>(Tips & Tricks are italicized and in parentheses)</i>
Disclaimer:	This software should not be construed as providing trading or investment recommendations. Any decisions you make using this software are done solely at your own risk. The risk of loss in trading can be substantial. Leveraged trading such as through options, futures, and forex exposes traders to heightened risk. Furthermore, traders may sustain losses greater than their investments, regardless of which asset classes are traded. Before trading, carefully consider the inherent risks in light of your financial condition.

Prerequisite Installation

1. TradeStation 9.1 (Update 13) or later
2. TradeStation data subscriptions for symbols input to the indicator.

Where to start after successful installation

1. Launch the Strategy Network demonstration workspace.
 - a. Note that it contains one chart with a custom composite index of SPY, IWM, QQQ, and DIA.
 - b. Further note the two instances of the indicator in the second chart. There it shows how the indicator can be used to create custom currency pairs by combining EURUSD and AUDUSD. The results are compared to the existing currency pair, EURAUD, to show the match along with calculated differences.
2. As an exercise, open a chart and insert the !QCL.MyIndex indicator. By simply removing comments from the default inputs you can reproduce the Dow Jones Industrial Average (for the 2010-07-02 construction).
3. First go to the Symbols input and remove “Symbol {DJIA: ” from the front of the input, and “}” from the end, leaving double quotes around the comma-separated string.
4. Next go to the MultipliersConcatStr input and remove the “1” {DJIA: ” from the front of the input, and “}” from the end, leaving double quotes around the comma-separated string.
5. Finally go to the CompositeMultiplier input and remove everything but 7.56833298376465. And press OK. It may take a few moments. The end result

- should be a plot of the Dow Jones Industrial Average. If you are entitled for the \$INDU, make it the Data1 chart symbol and compare.
- Build your own custom Index with !QCL.MyIndex. (If you have Microsoft Excel, you may want to use [!QCL.MyIndex_InputsTool.xls](#) with included macros to build comma-separated input strings. Download and save your copy using the link.)
 - More information is available at [QCLsolutions.com > Support > Add-Ons > MyIndex](#).

How to use

- Insert the indicator to Chart or Grid application in TradeStation 9.0.
- Plan your component and composite calculations:

Component = ((Symbol Close) ^ Power) * Multiplier + BaseValue

Components Combined = Component1 {operator1: +,-,*,/} Component2 {operator2}
Component3 ...

Composite = ((Components Combined) ^ CompositePower)) * CompositeMultiplier +
CompositeBaseValue

3. Setup Inputs:

<u>Inputs</u>	<u>Demo Values</u>	<u>Notes</u>
Notes	"Gold Index 1"	Notes for the user to keep relative to the current instantiation of the indicator. Shows up as the first input in charts when displayed in the status line.
ShowIndexComponents	FALSE	True if the user wants to see plots of the symbols used in the creation of the custom index.
ZeroIndexToStartBar	FALSE	True if when reach real time, the user wants to force the index value to zero. This helps one see real-time changes in index value.
ZeroAdjHistoryToStart	FALSE	Used with ZeroIndexToStartBar. True if the user also wants to back adjust historical plots relative to an initial real-time zero value for the index.
NegOrZeroSymbValuesOK	FALSE	False is recommended. True if the user wants to allow the index calculations to be affected by component symbols with negative or zero price values.
DaysBackToShow	0	0 to match the range of the TradeStation window, or the user can specify a number of days of history.
ShowStartAndZeroLine	TRUE	Used with ZeroIndexToStartBar. True if the user wants to show the bar on which real-time starts (with a vertical line) and to use a zero line (horizontal) with indicator plots as a reference for real-time changes.

StartMarkColor	White	Used with ZeroIndexToStartBar and ShowStartAndZeroLine. The color of the vertical line used to mark the start of real-time.
StartMarkStyle	Tool_Dotted	Used with ZeroIndexToStartBar and ShowStartAndZeroLine. The style of the vertical line used to mark the start of real-time. Can be Tool_Solid, Tool_Dotted, Tool_Dashed, Tool_Dashed2, or Tool_Dashed3.
StartMarkWidth_0to6	0	Used with ZeroIndexToStartBar and ShowStartAndZeroLine. It specifies the width of the vertical line used to mark the start of real-time. The input can be an integer from 0 to 6.
ZeroLineColor	DarkGray	Used with ZeroIndexToStartBar and ShowStartAndZeroLine. The vertical line color used to mark the start of real-time.
Symbols	"GOLD,GLD"	A comma separated string list of symbols to be used in creation of the custom index.
ShowAlsoOpen	TRUE	The indicator uses Closing bar values by default. Setting this input to True, also evaluates and displays values based on bar Opens.
ShowAlsoHigh	TRUE	The indicator uses Closing bar values by default. Setting this input to True, also evaluates and displays values based on bar Highs.
ShowAlsoLow	TRUE	The indicator uses Closing bar values by default. Setting this input to True, also evaluates and displays values based on bar Lows.
ShowAlsoVolume_012	2	Evaluates and displays composite calculations of volume. 0 disables, 1 enables with no rounding of volume, 2 enables with rounding for 3 rd Party data.
PowersConcatStr	"1,1"	A comma separated string list of powers to be used with price data for each corresponding symbol in Symbols.
MultipliersConcatStr	"1,1"	A comma separated string list of multipliers to be used with price data for each corresponding symbol in Symbols.
BaseValueConcatStr	"0,0"	A comma separated string list of base values (added to price * multiplier) used with price data for each corresponding symbol in Symbols. If there is no corresponding value, then 0 is assumed.
OperatorsConcatStr	"+"	A comma separated string list of operators to be used with consecutive symbol terms.

CompositePower	1	Each symbols price is adjusted by the formula (Multiplier * Price + BaseValue). The adjusted prices are added then raised by CompositePower.
CompositeMultiplier	1	Each symbols price is adjusted by the formula (Multiplier * Price + BaseValue). The adjusted prices are added then multiplied by CompositeMultiplier.
CompositeBaseValue	0	Composite prices (that have been multiplied by CompositeMultiplier) then have CompositeBaseValue added to complete the composite Index calculation.
SessionName	"24 Hour"	The session type to use with each symbol data stream used in the Index calculation. A blank string "" is interpreted as the default session of the PriceSeriesProvider object.
VolumeBars	FALSE	Set TRUE if using with Volume-Interval bars.
DisplacePlotBarsBack	0	A number of bars to displace plots backwards. Negative values plot bars forward within the limitations of available forward display space.
HiAlertLimit	250	A price limit value for the Index calculation that triggers an Alert when it is Crossed Over, on the last bar.
LoAlertLimit	200	A price limit value for the Index calculation that triggers an Alert when it is Crossed Under, on the last bar.
WriteToFile_FullPath	"C:\Temp\GoldIndex1.csv"	A full file path for recording the composite Index calculations by Date and Time. This data can be used with TradeStation as 3rd-Party data. If the input is a blank string (""), or too short, there is no attempt to write data to file.
AvoidMultiNtsData	TRUE	Enables MyIndex to detect requests for multiple non-linear time series data. When detected, returns 0 values and notifies of Error in a text plot, or raises a RunTimeError event if DoRaiseEventOnError is TRUE.
DoRaiseEventOnError	FALSE	If set to TRUE, when an error situation is detected like the multiple non-linear time series data request error, MyIndex will raise a RunTimeError event and be disabled.

4. Setup Alerts, Style, Color, and Scaling as with other TradeStation indicators.
5. Use from 1 to 100 input symbols. The maximum number of symbols that the indicator can realistically process is 100.
6. Note that there are plots for the Custom Index and up to 47 symbol components.
(When ShowIndexComponents is True, each Symbol has a plot of its name and

- then value. Due to a legacy limitation of TradeStation charts, with chart tip windows, symbol name plots are only visible on the last bar.)*
7. Note that data provider objects are new and powerful EasyLanguage components. There can be issues using indicators that leverage them. Make sure your platform is online and entitled for the necessary data. *If an indicator fails to plot for an extensive period, you may want to refresh the related TradeStation window. In some cases you may need to reboot the TradeStation platform.*
 8. Note that in RadarScreen, the indicator may initially appear to be in an error state with an **E** in the lower-left corner. The TradeStation message is “The study tried to reference data before it was available”. *The RadarScreen instance will work once data arrives, so simply give it time.* If it takes too long then you may want to refresh the symbol.

Demonstration EasyLanguage

Below is an EasyLanguage example of using mMyIndexEX:

```
{ mMyIndexEX demo for instructions }
{ October, 9 2012: http://en.wikipedia.org/wiki/DJIA\_divisor }
```

Inputs:

```
    string Symbols(
"AA,AXP,BA,BAC,CAT,CSCO,CVX,DD,DIS,GE,HD,HPQ,IBM,INTC,JNJ,JPM,KO,MCD,MMM,M
RK,MSFT,PFE,PG,T,TRV,UNH,UTX,VZ,WMT,XOM"),
    string PowersConcatStr("1"),
    string MultipliersConcatStr("1"),
    string BaseValueConcatStr("0"),
    string OperatorsConcatStr("+"),
    double CompositePower(1),
    double CompositeMultiplier(7.679543051215), //reciprocal of divisor
    double CompositeBaseValue(0),
    string SessionName(""),
    bool VolumeBars(False);
```

Variables:

```
    double oOpen(0),
    double oHigh(0),
    double oLow(0),
    int64 oVolume(0),
    string oState(""),
    string oNoteOnError("");
```

```
Value0 = mMyIndexEX(
    Symbols,
    PowersConcatStr,
```

```
MultipliersConcatStr,  
BaseValueConcatStr,  
OperatorsConcatStr,  
CompositePower,  
CompositeMultiplier,  
CompositeBaseValue,  
SessionName,  
VolumeBars,  
oOpen,  
oHigh,  
oLow,  
oVolume,  
oState,  
oNoteOnError);
```

```
If oNoteOnError = "" then Begin  
//Same as//If mStringContains(oState, "Loaded", True) then Begin  
    Plot1(oOpen, "Open");  
    Plot2(oHigh, "High");  
    Plot3(oLow, "Low");  
    Plot4(Value0, "Close");  
    Plot5(NumToStr(oVolume, 0), "Volume");  
End;  
Plot6(oState, "State");  
Plot7(oNoteOnError, "NoteOnError");
```

For additional examples visit us at: [MyIndex at QCLsolutions.com](http://MyIndex.at.QCLsolutions.com)