



User Guide

© 2004-2005 FMR Corp. All rights reserved.

Index-Lab User Guide

by FMR Corp.

Revised: Friday, April 15, 2005

Index-Lab User Guide

© 2004-2005 FMR Corp. All rights reserved.

No parts of this work may be reproduced in any form or by any means - graphic, electronic, or mechanical, including photocopying, recording, taping, or information storage and retrieval systems - without the written permission of the publisher.

Third party trademarks and service marks are the property of their respective owners.

While every precaution has been taken in the preparation of this document, the publisher and the author assume no responsibility for errors or omissions, or for damages resulting from the use or misuse of information contained in this document or from the use or misuse of programs and source code that may accompany it. In no event shall the publisher and the author be liable for any loss of profit or any other commercial damage caused or alleged to have been caused directly or indirectly by this document.

Printed: Friday, April 15, 2005

Special thanks to:

Wealth-Lab's great on-line community whose comments have helped make this manual more useful for veteran and new users alike.

EC Software, whose product HELP & MANUAL printed this document.

Table of Contents

Foreword	0
Part I Introduction	2
1 Index-Lab Overview	2
2 Wealth-Lab Online Community	3
Part II Using Index-Lab	4
1 Creating and Managing IndexScripts	4
2 Creating Indices with OHLC/V Values	7
3 Selecting your WatchLists	8
4 Updating your Indices and Aggregate Indicators	9
5 Using Index-Lab in Wealth-Lab Pro	10
Part III Sample IndexScripts	13
1 Advancing minus Declining Issues	13
2 Advance/Decline Line	14
3 Daily Change	15
4 MACDBull/MACDBear	16
5 RSI14	17
Index	18

1 Introduction

1.1 Index-Lab Overview

What is Index-Lab?


Index-Lab is an Add-On module for Wealth-Lab Pro. Using Index-Lab you can easily create and update your own unique Market Indices and Aggregate Indicators. Here are some examples of what you can do with Index-Lab:

- Create a Market Index that tracks the average gain/loss of any group of symbols that you select.
- Create custom Advance/Decline Lines to measure broad market performance, and create other Market Breadth indicators.
- Create Aggregate Indicators, such as the average RSI for all symbols in the Index. Take action when the Aggregate Indicator reaches oversold or overbought levels.
- Capture the "Breath" of the market with Aggregate Indicators like Tsokakis MACDBull and Tsokakis MACDBear.

How Does Index-Lab Work?

Index-Lab uses **IndexScripts** which define how to create the Index or Aggregate Indicator. IndexScripts are simple scripts written in our WealthScript scripting language. You work with IndexScripts using the **IndexScript Editor**. For more information on the WealthScript language, please consult the WealthScript Language Guide documentation that comes with Wealth-Lab Pro.

Once your IndexScripts are created, you can assign them to one or more **WatchLists** using the WatchList Selection Interface. This assignment determines which Indices and Aggregate Indicators are created for which WatchLists.

The last step is to start the [Update](#)  process to generate the Indices and Aggregate Indicators. Each IndexScript/WatchList pair that was selected results in a new symbol being created.

Index-Lab Creates a New DataSource

It's important to note that Index-Lab creates a new Wealth-Lab Pro DataSource called "Index-Lab". All of the indices and aggregate indicators that Index-Lab creates become new symbols in this DataSource. These symbols can be accessed in any ChartScript via that standard **SetPrimarySeries**, **GetExternalSeries**, or **PlotSymbol** calls. You must restart Wealth-Lab Pro to see new symbols that are created by Index-Lab.

1.2 Wealth-Lab Online Community

Be sure to visit our web site, www.wealth-lab.com¹. We host a dedicated trading system development community. The web site contains many useful features ...

[Knowledge Base](#)

We publish articles related to technical analysis and techniques for using Wealth-Lab products.

[ChartScripts](#)

You can actually develop and back test ChartScripts on site, and publish your systems for peer review and feedback. Also, try the ChartScripts published by other members to get fresh ideas.

[DynaCharts](#)

The DynaCharts tool lets you create customized charts containing a wide array of technical indicators by just entering your selections in a web-based form. You can then generate the WealthScript code for your creation.

[WatchList Scans](#)

You can scan a WatchList of up to 100 symbols. This feature is great if you need to get your daily signals from an internet cafe when you're on holiday.

[\\$imulator](#)

The on-site \$imulator lets you run simulations using WatchLists of up to 100 symbols.

[Discussion Forum](#)

Our forums are dedicated to trading system development. Use them to exchange ideas and get help.

[Function Reference](#)

Our on-site Function Reference is fully dynamic. When you view the code for a ChartScript, each function is linked to its entry in the Reference. You can add your own comments and URLs to any Reference entry.

¹ Legal Disclaimer: Please note that Wealth-Lab.com is a non-Fidelity site. Fidelity has not been involved in the preparation of the content supplied at this third party site and does not guarantee or assume any responsibility for its content.

2 Using Index-Lab

2.1 Creating and Managing IndexScripts

What is an IndexScript?

An **IndexScript** is a fragment of WealthScript code that **returns a value** used to calculate the index for **one symbol in the list**. The IndexScript is automatically passed a **Bar** variable. The script should calculate and return the index value for that bar number. The IndexScript should assign the return value to the **Result** variable, as if it were the body of a WealthScript function.

Index-Lab contains a user-interface to perform **post processing** on the individual values returned by your Index-Script. You can, for example, select to average or sum all of the individual values.

Basic Example

The most simple IndexScript possible returns the average of all closing prices for the symbols in the list. The IndexScript would be coded like this:

```
Result := PriceClose( Bar );
```

The script simply returns the closing price of each symbol. In the post processing phase (see below) you can tell Index-Lab to average all of the values to produce the final index.

More Complex Example

The following example calculates Advancing minus Declining issues. The script simply assigns a return value of 1 if the closing price was greater than the previous bar's close, and a -1 if it was lower. After all of the symbol are processed, Index-Lab will sum up these ones and negative ones to arrive at Advancing minus Declining Issues.

```
if PriceClose( Bar ) > PriceClose( Bar - 1 ) then
  Result := 1
else if PriceClose( Bar ) < PriceClose( Bar - 1 ) then
  Result := -1
else
  Result := 0;
```

IndexScript Symbol

Each IndexScript must be assigned a unique **Symbol**. This symbol is used when creating the final symbols used to refer to your Indices and Aggregate Indicators. Each is assigned a symbol that is a combination of the **IndexScript Symbol** and the **WatchList Symbol** (see the [WatchList Selection](#)^[8] topic).

Index-Lab creates the combination symbols by appending the IndexScript symbol and the WatchList symbols with an underscore. For example, assume that the **Advance Decline Line** IndexScript has a Symbol of "AD". If you generate the Advance Decline Line for a WatchList with a symbol of "DJ", the final symbol becomes DJ_AD.

IndexScript Editor

Use the **IndexScript Editor** to create, modify and delete IndexScripts. During the editing process you can press the **Check for Errors** button (or menu item, or press F5) to test the script for errors. Index-Lab comes with a number of example IndexScripts that you can use as-is, or modify to build your own Indices and Aggregate Indicators.

Post Processing

The IndexScript calculates the value that each symbol in the list will contribute to the overall Index or Aggregate Indicator. Once all of these values are calculated, Index-Lab performs **post processing** on the values based on the settings that you specify in the **Post Processing** section (under the main code editor).

Index=Sum

Index-Lab sums the individual return values for each symbol to compute the final value.

Index=Average

Index-Lab averages the individual return values for each symbol to compute the final value. Index-Lab uses only the symbols that contained data for a given bar when determining the number of symbols to base the average on. This results in correct index values even if some symbols in the list did not trade on a specific date.

Add to Previous Index Value

This option instructs Index-Lab to add the resulting value to the resulting value from the previous bar.

Multiply by Previous Index Value

This option instructs Index-Lab to multiply the resulting value to the resulting value from the previous bar. If the previous value is zero Index-Lab does not perform the multiplication.

Divide by Previous Index Value

This option instructs Index-Lab to divide the resulting value by the resulting value from the previous bar. If the previous value is zero Index-Lab does not perform the division.

Divide into Previous Index Value

This option instructs Index-Lab to divide the resulting value into the resulting value from the previous bar. This operation is performed only if both values are non-zero.

Lead Bars

Specifies the number of bars back that the script will look in the chart. For example, if you are using a 30 period RSI in your Index-Script you should set Lead Bars to *at least* 30. (For indicators, that use Wilder smoothing, like RSI, a rule of thumb is to allow 2.5 times the period to allow the indicator to stabilize sufficiently.) You should set Lead Bars to at least 31 if you are referencing the previous result in your post processing selection.

Start Index Value

Specifies the initial value to assign to the Index. This value is used only if the Index is set to perform an operation on its previous value. For example, you might start an index at a value of 100, and then multiply this value by the average percentage change going forward.

Multiply Final Result by N

Multiplies the final Index value by the constant specified in the edit field. Your IndexScript might calculate average percentage gain, for example, and you can multiply the final result by 100.

2.2 Creating Indices with OHLC/V Values

Complex Indices

You can use Index-Lab to create indices with complete open, high, low, close and volume values. Just as a normal IndexScript uses the **Result** variable to assign the index value, you use other special variables to assign open, high, low and volume values.

Index Value	Variable Name
Open	ResultOpen
High	ResultHigh
Low	ResultLow
Close	Result (note: not ResultClose!)
Volume	ResultVolume

Use all Variables!

If you want to create an index that contains OHLC/V values, be sure to assign values to **all of the result variables**. If you neglect to assign a result, it will assume a zero value.

Example

The following example is the most basic type of index that uses OHLC/V values. It simply averages the prices and volume for all symbols in the list. In post processing, be sure to select the option to Average the values.

```
Result := PriceClose( Bar );
ResultOpen := PriceOpen( Bar );
ResultHigh := PriceHigh( Bar );
ResultLow := PriceLow( Bar );
ResultVolume := Volume( Bar );
```

Index-Lab Override

If a complex index contains a high value that is less than the open or close, or a low value that is greater than the open or close, Index-Lab will adjust the high or low value accordingly.

2.3 Selecting your WatchLists

WatchList Selection

The WatchList Selection interface is a grid that displays all of your Wealth-Lab Pro WatchLists as rows and all of your IndexScripts as columns. You simply click the grid cells when you want to have an Index or Aggregate Indicator generated for a specific WatchList. If you want to generate all of the available Indices/Indicators for a WatchList, double click the WatchList name and all of the grid cells will be selected. Double click again to clear out the selections.

WatchList Symbol

In order to select a WatchList you must first provide a **Symbol** for it. You do this by clicking twice on the Symbol column of the WatchList and typing in the symbol (the column will originally contain the word "<NONE>"). Also, trying to click a grid cell for a WatchList that does not have a Symbol will automatically put you in Symbol entry mode.

The WatchList Symbol, along with the IndexScript symbol, is used to create the final symbol for the Index or Aggregate Indicator. The final symbol is composed of the WatchList symbol appended with the IndexScript symbol, separated by the underscore (_) character.

Example

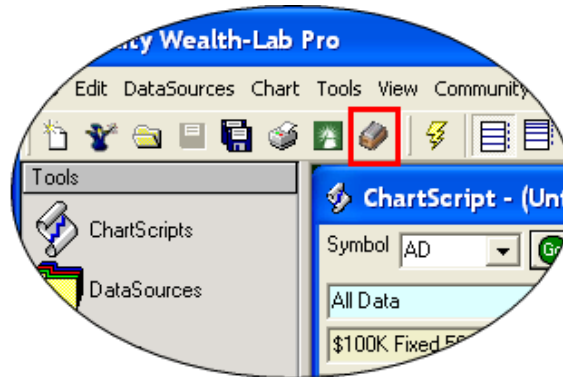
IndexScript = Advance Decline Line (**Symbol**=AD)

WatchList = Dow 30 (**Symbol**=DJ)

Resulting Symbol = DJ_AD

2.4 Updating your Indices and Aggregate Indicators

After installing Index-Lab, its icon will appear in Wealth-Lab Pro's File tool bar. Click this icon to launch Index-Lab and use one of the options below for updating.



You can launch Index-Lab from within Wealth-Lab Pro by clicking the icon in the File toolbar.

Updating

The **Update Indices** interface is where you perform your daily update. You should update the Indices each day after your data update is complete. Click the "Click here to Update all Indices" button to proceed. All of the Indices and Aggregate Indicators that you set up in the WatchList Selection interface will be generated, one at a time.

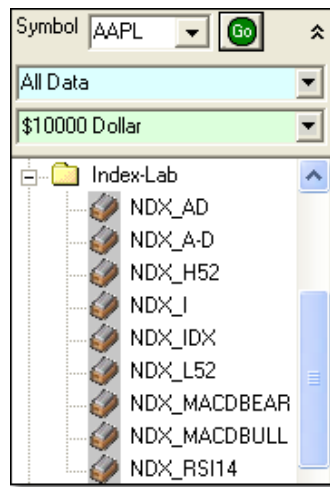
One-Click Updating

You can also click the ⚡ Update button on the toolbar (or menu item, or press F9) to start the Update procedure.

2.5 Using Index-Lab in Wealth-Lab Pro

Index-Lab DataSource

Index-Lab creates a new Wealth-Lab Pro **DataSource** where all of the Indices and Aggregate Indicators are stored. The DataSource is called **Index-Lab**. The DataSource contains a symbol for each of the IndexScript/WatchList pairs that you defined in the [Selection](#) interface.



You can access these symbols in any ChartScript by using the **GetExternalSeries** WealthScript function.

Example

The following script accesses and plots several Index-Lab symbols on the chart.

```

var NDX_AD, NDX_RSI14, NDX_MACDBEAR, NDX_MACDBULL: integer;
var ADPane, RSIPane, MACDPane: integer;

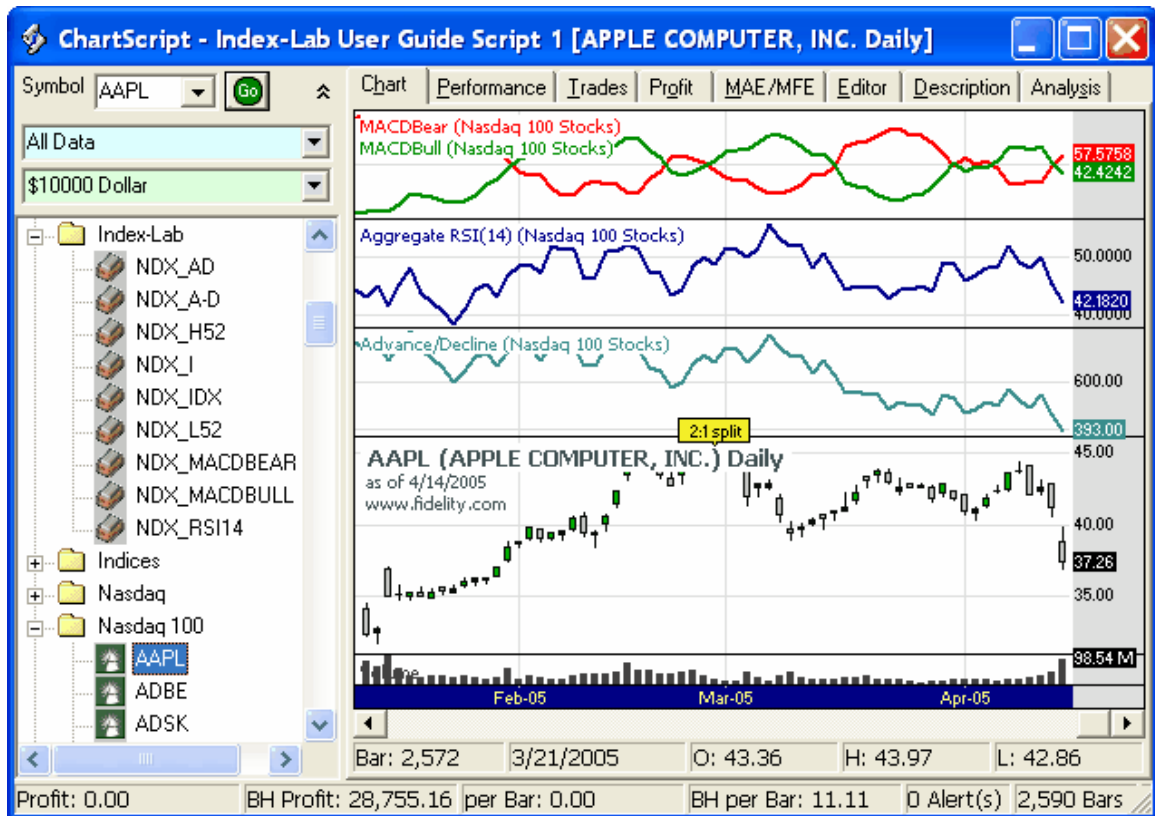
{ Create the panes }
ADPane := CreatePane( 100, true, true );
RSIPane := CreatePane( 100, true, true );
MACDPane := CreatePane( 100, true, true );

{ Get Index-Lab Symbols for the Nasdaq 100 }
NDX_AD := GetExternalSeries( 'NDX_AD', #Close );
NDX_RSI14 := GetExternalSeries( 'NDX_RSI14', #Close );
NDX_MACDBEAR := GetExternalSeries( 'NDX_MACDBEAR', #Close );
NDX_MACDBULL := GetExternalSeries( 'NDX_MACDBULL', #Close );

{ Plot them }
PlotSeriesLabel( NDX_AD, ADPane, #Teal, #Thick, 'Advance/Decline (Nasdaq
100 Stocks)' );
PlotSeriesLabel( NDX_RSI14, RSIPane, #Navy, #Thick, 'Aggregate RSI(14)
(Nasdaq 100 Stocks)' );
PlotSeriesLabel( NDX_MACDBEAR, MACDPane, #Red, #Thick, 'MACDBear (Nasdaq
100 Stocks)' );
PlotSeriesLabel( NDX_MACDBULL, MACDPane, #Green, #Thick, 'MACDBull

```

```
(Nasdaq 100 Stocks)' );
```



Example

This example calculates a **McClellan Oscillator** type indicator for the Nasdaq 100 WatchList. The McClellan Oscillator is based on the Advancing - Declining Issues IndexScript. It is defined as the 39 bar EMA of A-D minus the 19 bar EMA of A-D. After running the script on symbol AAPL we dragged a 30 bar Simple Moving Average (SMA) from the Indicator List on top of the Oscillator, to create a smoothed version.

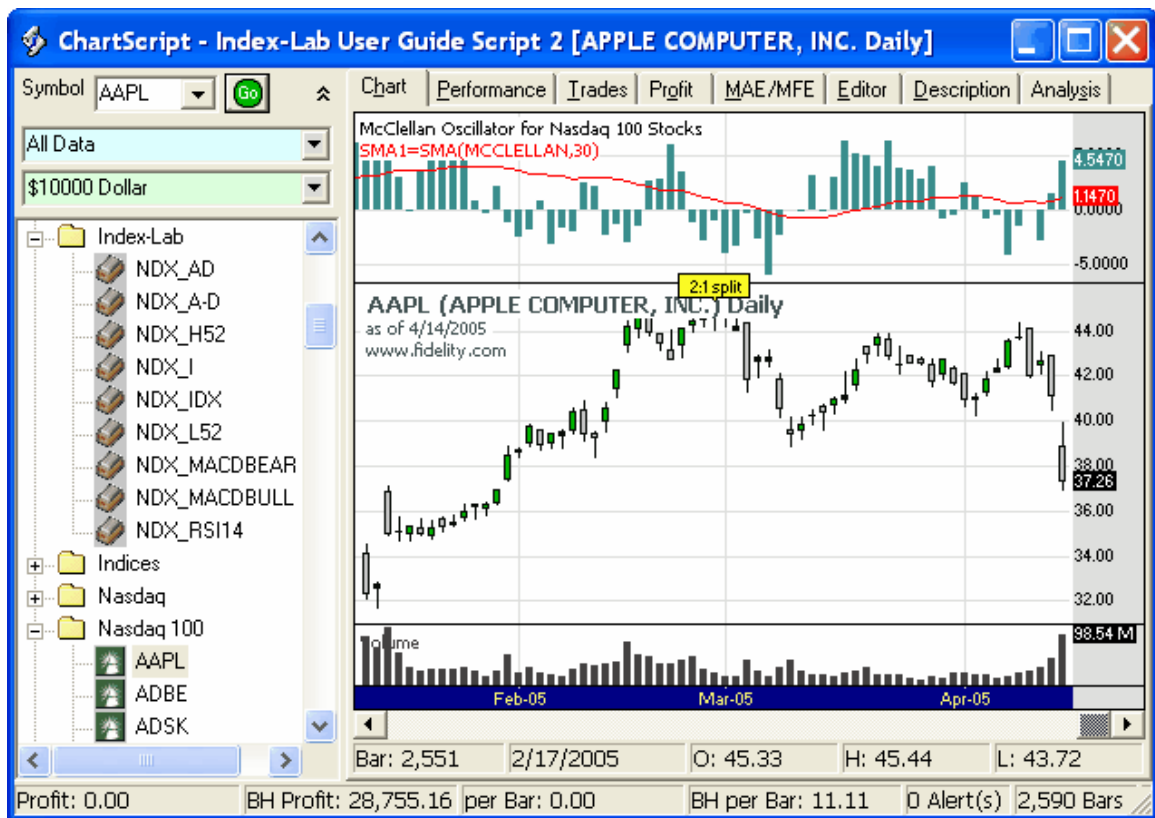
```
var MCPANE, AD, EMA39, EMA19, MCCLELLAN: integer;

{ Create Pane }
MCPane := CreatePane( 100, true, true );

{ Get Index-Lab Symbols for Dow 30 Advancing - Declining Issues }
AD := GetExternalSeries( 'NDX_A-D', #Close );

{ Calculate McClellan Oscillator }
EMA39 := EMASeries( AD, 39 );
EMA19 := EMASeries( AD, 19 );
McClellan := SubtractSeries( EMA39, EMA19 );

{ Plot McClellan Oscillator }
PlotSeries( McClellan, MCPane, #Teal, #ThickHist );
DrawLabel( 'McClellan Oscillator for Nasdaq 100 Stocks', MCPane );
```



3 Sample IndexScripts

3.1 Advancing minus Declining Issues

IndexScript Code

```
if PriceClose( Bar ) > PriceClose( Bar - 1 ) then
  Result := 1
else if PriceClose( Bar ) < PriceClose( Bar - 1 ) then
  Result := -1
else
  Result := 0;
```

Post Processing

```
Index = Sum
Lead Bars = 1
```

Description

Produces the Advancing Issues minus Declining Issues breadth indicator. This is the basis for several more complex breadth measurements, including the McClellan Oscillator.

Note that this entry is almost exactly the same as the Advance Decline Line IndexScript. The only difference is that the Advance Decline Line adds the result to the previous Index value, resulting in a cumulative Index, while this returns the raw daily values.

3.2 Advance/Decline Line

IndexScript Code

```
if PriceClose( Bar ) > PriceClose( Bar - 1 ) then
  Result := 1
else if PriceClose( Bar ) < PriceClose( Bar - 1 ) then
  Result := -1
else
  Result := 0;
```

Post Processing

```
Index = Sum
Add to Previous Index Value
Lead Bars = 1
```

Description

Creates the Advance/Decline line, which is the cumulated sum of Advancing Issues minus Declining Issues. The Advance/Decline line is one of the most widely used breadth indicators in technical analysis.

3.3 Daily Change

IndexScript Code

```
Result := ( PriceClose( Bar ) - PriceClose( Bar - 1 ) ) / PriceClose( Bar  
- 1 ) + 1;
```

Post Processing

```
Index = Average  
Multiply by Previous Index Value  
Lead Bars = 1  
Start Index Value = 100
```

Description

Creates a custom index with a equal dollar weighting scheme rebalanced daily. The Index starts at 100. For each bar, the IndexScript takes the average daily percentage change and multiplies it by the previous value of the Index.

3.4 MACDBull/MACDBear

MACDBull and MACDBear are two "Breath Indicators" created by Dimitris Tsokakis.

IndexScript Code (Tsokakis MACDBull Shown)

```
var Signal: integer;
Signal := EMASeries( MACDSeries( #Close ), 9 );
if MACD( Bar, #Close ) > GetSeriesValue( Bar, Signal ) then
    Result := 1
else
    Result := 0;
```

Post Processing

```
Index = Average
Lead Bars = 30
Multiply Index Value by 100
```

Description

Tsokakis MACDBull is an Aggregate Indicator shows the percentage of symbols whose MACD is above the Signal Line (bullish). Tsokakis MACDBear is the inverse of MACDBull, and shows the percentage of symbols whose MACD is below the Signal Line.

3.5 RSI14

IndexScript Code

```
Result := RSI( Bar, #Close, 14 );
```

Post Processing

```
Index = Average  
Lead Bars = 14
```

Description

This is an Aggregate RSI (Relative Strength Index) Indicator. It is simply the average 14 bar RSI value for all symbols in the WatchList.

Index

- A -

Advance/Decline Line 14
Advancing minus Declining Issues 13

- C -

Close 7
Complex Indices 7

- D -

Daily Change 15

- E -

Editor 4
Examples 10

- H -

High 7

- I -

Index-Lab Update 9
IndexScript 4

- K -

Knowledge Base 3

- L -

Lead Bars 4
Low 7

- M -

MACDBear 16
MACDBull 16

- O -

OHLV 7
Open 7
Overview 2

- P -

Post Processing 4

- R -

RSI14 17

- S -

Samples 10
Selection 8
Start Index Value 4
Symbol 4, 8

- U -

Update 9
Updating 9

- V -

Volume 7

- W -

WatchList 8
Wealth-Lab Pro 10
Wealth-Lab.com 3