Index Lab

Understand how to create your own Index Script.

Introductions

The intention of this document is to describe the Index-Lab API. Users can create there own Index Scripts with the help of this document. Existing indexes, already plugged-in with Wealth-Lab Pro, are part of .net library "WealthLab.IndexDefinitions" and each index script is descendent of an abstract class "*IndexDefinition*". This abstract class is part of class library WealthLab.dll.

Building your own Index

You can create your own index script definition by using any .Net 2.0 development platform and any .Net language. All the new Index Script (s) need(s) to be created in a separate .Net class library (DLL). You can combine your entire newly created index Script(s) into one .Net Class Library. When this DLL is being placed in Wealth-Lab Pro installation folder, Wealth-Lab Pro will discover and include these index scripts to the list of existing index scripts definitions.

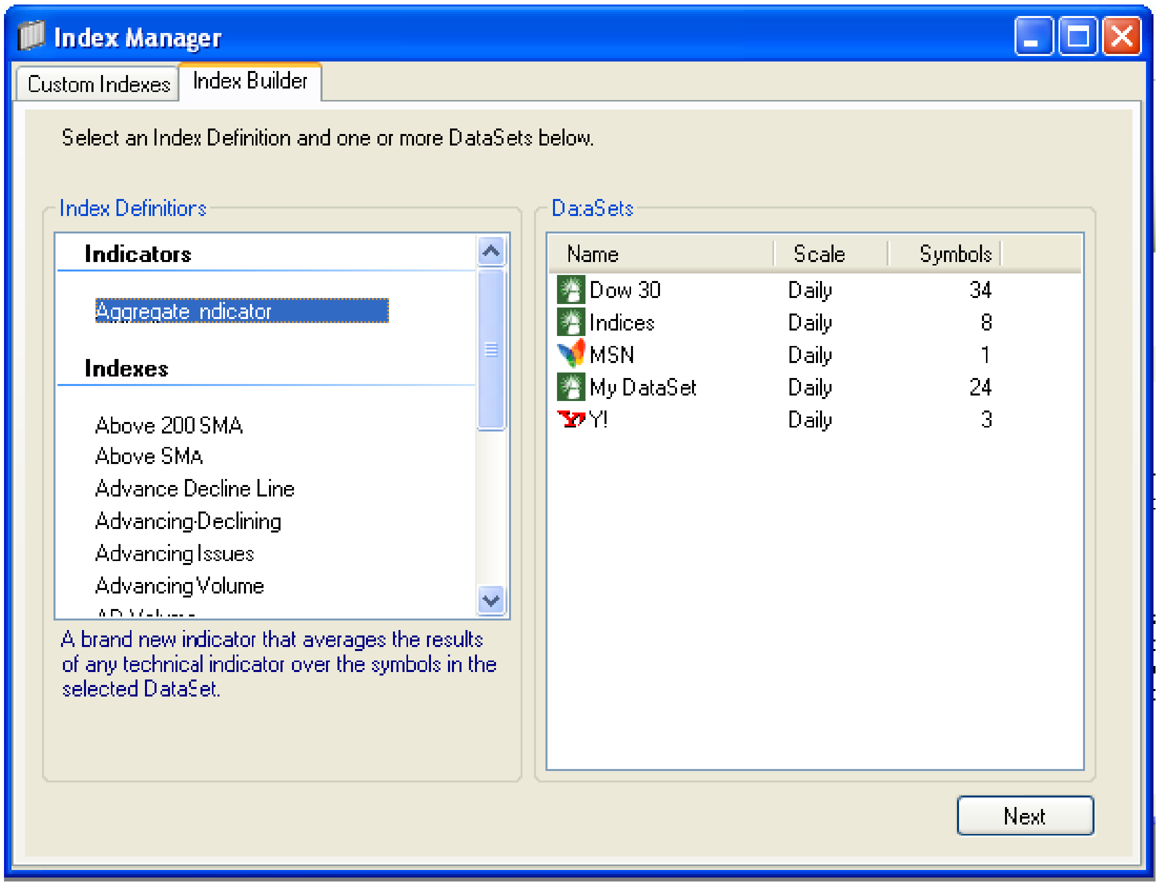
Steps to create your own index definition

1. Create a Class Library project in Visual Studio (or some other .NET development platform).
2. Add reference to the **WealthLab.dll** assembly in your project's References section. This assembly contains classes in the WealthLab namespace.

3. Create a new class. This class should be derived from '*IndexDefinition*'. This, IndexDefinition, class will contain the methods and properties, which will be called by Wealth-Lab Pro to process your Index Script. *Please note that, you can create as many index Scripts as you want, by adding classed to this library or you may create a separate library for each index Script.*

To test newly created Index, do the following

1. Drop your new **class library**, created above, into wealth-lab pro installation folder.
2. Run Wealth-Lab Pro application.
3. Press Ctrl+Alt+I, you will see **Index Manager Tool**. Go to **Index builder** tab and you will see your index Script under index definition section.



Select an Index Definition and one or more DataSets below.

Index Definitions

Indicators

Aggregate Indicator

Indexes

Above 200 SMA
Above SMA
Advance Decline Line
Advancing-Declining
Advancing Issues
Advancing Volume

A brand new indicator that averages the results of any technical indicator over the symbols in the selected DataSet.

DataSets

Name	Scale	Symbols
Dow 30	Daily	34
Indices	Daily	8
MSN	Daily	1
My DataSet	Daily	24
Y!	Daily	3

IndexDefinition Abstract Class

Your new Index Script will be descendent of IndexDefiniation abstract class, following are mandatory and optional methods and properties that it needs to override.

Methods

`public abstract void SetIndexValues(List<Bars> bars, List<int> barNumbers, DateTime dt, Bars indexToDate);` This method is the core of the Index processing - it must be overridden to provide a return value based on the list of Bars objects and associated index values, and the index values to date
`public virtual bool ValidateUserInput(ref string errMsg)` Validates the user input to the parameter user interface. Returns true if the validation succeeds; otherwise, returns false.
`public virtual void ClearUserInterface()` If the Index Script supports custom parameters, this method is called to clear the user interface form

Properties

`public abstract string FriendlyName` Friendly name for the index. Specify the name that you want to display under index definition list. For example Above 200 SMA, Above SMA are the Friendly names their respective scripts
`public abstract string Description` Description of the index. Specify the description that you would like to see for your script. For example below is the description for Aggregate Indicators "*`A brand new indicator that averages the results of any technical indicator over the symbols in the selected DataSet.`*"
`public abstract string Prefix` This will be used when creating the symbol from this script.
`public virtual bool SupportsParameters` Specifies your script needs custom parameters or not.
`public virtual bool NeedsSeparateUIForParameters` Specifies your script needs separate UI for your specified custom parameters or not. *To set this as 'true', you have to specify "`SupportsParameters`" as true.*
`public virtual UserControl ParameterUserInterface` If the Index Script supports custom parameters, it returns a user interface where the parameters can be entered
`public virtual string ParameterString` If the Index Script supports custom parameters, it must express them as a single string based on the input into theparameters user interface referenced above.

Fidelity Brokerage Services LLC, Member NYSE, SIPC
900 Salem Street, Smithfield, RI 02917