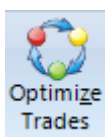


Optimize Trades



Documentation of the Optimize Trades feature will continue with the [DYO - Trading System](#) example which initiates a Long position with a Buy Stop at an upper Bollinger band, and seeks to exit on a profit objective. The trade is protected by a trailing stop at the lower Bollinger band level.

Though the example looks promising with favorable trade results, one does not know if the parameters chosen are optimal. For example, would more profit be achieved by trying to scalp a larger move? Or, should the Bollinger bands used for entry and exit be closer to the price action, or farther away?

Ensign has a powerful tool that can be used to research these questions. Click the Optimize Trades button on the Setup ribbon to show the following form.

The screenshot shows the 'Optimizer' software window with the 'Setup' tab selected. The interface includes a dropdown menu for the 'Optimizer control file' set to 'MACD Bollinger', and buttons for 'Save', 'Run Optimizer', and 'Clear Form'. Below this is a 'List of Layouts or Charts' area. To the right, there are checkboxes for 'Evaluate each layout or chart on the list' and 'Results print method'. A table defines three optimization loops: LoopA (MACD, 25 to 35, step 5), LoopB (BOL, 1.8 to 2.2, step 0.2), and LoopC (Scalp, 0.0020 to 0.0034, step 0.0002). Below the table are radio buttons for 'Optimize' (Trade Profits, Minimize Variable, Maximize Variable) and a dropdown for 'Assign Variable' set to '0'. At the bottom, there is a 'Load Template' dropdown set to 'JaysonOptimize', a checked 'Execute Script' checkbox, and a calculation '3 x 3 x 8 = 72 iterations'.

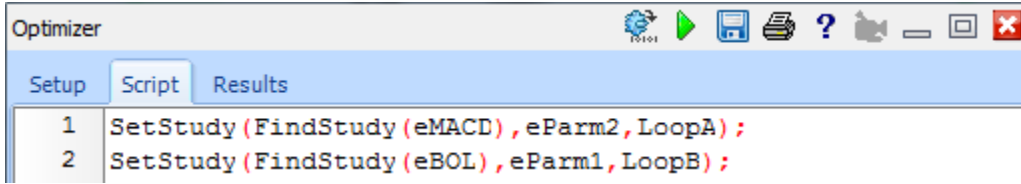
Iterations

The Setup tab is used to define the iterations that are to be tested. This example will vary the parameter for the 2nd MACD average from 25 to 35 in increments of 5. The width multiplier parameter for the Bollinger bands will be tested in the range of 1.8 to 2.2 in increments of 0.2. And the scalp objective will be varied from 20 pips to 34 pips in 2 pip increments. These 3 parameters are nested loops, where LoopC is the inner most loop.

The MACD loop tests 3 values. The BOL loop tests 3 values. And the Scalp loop tests 8 variations. The total number of combinations examined will be $3 \times 3 \times 8 = 72$.

Execute Script

This example will use a small ESPL script entered on the 2nd tab to place the loop values in the MACD and Bollinger studies. The box is checked so this script executes with each iteration. Variables named LoopA, LoopB and LoopC are available for use in the script.



```

1 SetStudy (FindStudy (eMACD) , eParm2, LoopA) ;
2 SetStudy (FindStudy (eBOL) , eParm1, LoopB) ;
  
```

The ESPL script for the Optimizer is entered on the 2nd tab of the form. The script for this example consists of 2 lines. Line 1 finds the MACD study and sets its 2nd parameter to the LoopA value. Line 2 finds the Bollinger study and sets its 1st parameter to the LoopB value. Refer to the ESPL.PDF manual for a list of study constants, and documentation of the SetStudy statement. The extent of your scripts will probably be like this 2 line example.

The LoopC value needed for the scalp objective is assigned to a Variable selected on the 1st tab. The example is placing the LoopC value in Variable [50].

	Title	From	To	Step	Assign Variable
<input checked="" type="checkbox"/> Enable LoopA	MACD	25	35	5	
<input checked="" type="checkbox"/> Enable LoopB	BOL	1.8	2.2	0.2	
<input checked="" type="checkbox"/> Enable LoopC	Scalp	0.0020	0.0034	0.0002	50

Since the LoopC iteration value will be in Variable [50], the DY0 script needs to be modified to use this variable value instead of having a fixed scalp objective of 30 pips. Row H of the DY0 is modified to add the variable value to the upper Bollinger band value.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Expression	32 Long Scalp	= [Bollinger Upper]	+	[50]	0
<div style="border: 1px solid black; padding: 2px;"> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> </div>					
A	Study	[MACD] := MACD.MACD value			
B	Study	[MACDave] := MACD.Average value			
C	Study	[Bollinger Upper] := BOL.1st Upper band			
D	Study	[Bollinger Lower] := BOL.1st Lower band			
E	Expression	[28] := ([MACD] > 0) AND ([MACD] > [MACDave])		ZON	
F	Flag	[Rule1b] := [E] goes True		ZON	
G	Action	if ([Rule1b] AND Position is Flat) then Buy Stop Price := [Bollinger Upper]			
H	Expression	[Long Scalp] := [Bollinger Upper] + [50]			

The Optimizer is now ready to do its recalculation of the system 72 times. Click the Run Optimizer button on the 1st tab. The view will change to show the 3rd tab which has a table of trade results.

Results

	Chart	MACD	BOL	Scalp	Win #	Loss #	Total #	% Win	\$ Win	\$ Loss	Profit	Ratio	Average
35	EUR_USD.2	30.00	1.8000	0.0032	32	24	56	57	5812	-2033	3780	2.8596	67.50
36	EUR_USD.2	30.00	1.8000	0.0034	31	25	56	55	4877	-2162	2715	2.2559	48.48
38	EUR_USD.2	30.00	2.0000	0.0020	32	27	59	54	5298	-2396	2903	2.2115	49.20
39	EUR_USD.2	30.00	2.0000	0.0022	31	27	58	53	5579	-2474	3105	2.2553	53.53
40	EUR_USD.2	30.00	2.0000	0.0024	30	26	56	54	5634	-2462	3172	2.2887	56.65
41	EUR_USD.2	30.00	2.0000	0.0026	28	26	54	52	5554	-2380	3174	2.3337	58.78
42	EUR_USD.2	30.00	2.0000	0.0028	28	25	53	53	5750	-2154	3595	2.6689	67.84
43	EUR_USD.2	30.00	2.0000	0.0030	28	24	52	54	5886	-1973	3913	2.9831	75.24
44	EUR_USD.2	30.00	2.0000	0.0032	27	24	51	53	5048	-2104	2944	2.3994	57.73
45	EUR_USD.2	30.00	2.0000	0.0034	27	24	51	53	5308	-2104	3204	2.5229	62.83
47	EUR_USD.2	30.00	2.2000	0.0020	30	24	54	56	5342	-2570	2771	2.0782	51.32

The table contains the trade summaries for the 72 tests made. The view shown above has been positioned to show the settings that generated the highest Profit, which is Row 43. For the test sample which spanned a 7 day period at the end of 2010, the optimal parameters would have been to use a 30 period average for the 2nd MACD average, a 2.0 multiplier for the Bollinger bands, and a scalp objective of 30 pips.

Another use for the table of results is to evaluate how sensitive a system's design might be the parameters that are used. For example, compare the results of rows 38 through 41. The scalp parameter is varied from 20 pips to 26 pips, but the Profit remained quite steady through that range of scalp values.

File of Optimal Parameters

The optimal parameters found for Loop A, Loop B, and Loop C are written to an ASCII file in the C:\Ensign10\Optimizer\ folder. The file name used is the chart's filename with a suffix of '.txt' appended. The file will contain the values for Loops A, B and C separated by commas.

Example filename: C:\Ensign10\Optimizer\EUR_USD.2.txt

Example content: 30.0000, 2.0000, 0.0030

A DY0 can read these values with a function. Function | Optimizer File Field(#2)

The #2 value should be 1, 2 or 3 to select the 1st, 2nd, or 3rd field to read from this file.

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4	Offset
Function	5 Scalp	= Optimizer File Field(#2)	2		0
		{			0
A	Function	[Scalp] := Optimizer File Field(2)			

Optimizer Running

An improvement to the design would be to make the DYO automatically adapt itself for when it is run normally, and for when it is controlled by the Optimizer. A row could have been added that conditionally sets [50] to 0.003 when the Optimizer is not running. The statement would be:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4
Action	50	= if ## then V := #2		0.003
		Not	{	Optimizer Running
A Action if (Not Optimizer Running) then [50] := 0.003				

Such a statement would permit the the DYO to be run normally or by the Optimizer. The Optimizer Running flag will be True when the Optimizer is running.

The next example shows how normal assignments for several Variables can be skipped when the Optimizer is running. Add an initialization DYO as the first DYO in a design, like this:

Category	Variable	Selection #1 & #3	Op. [#]	Selection #2 & #4
Branching	0	= if # then abort execution		Optimizer Running
A Branching if Optimizer Running then abort execution				
B	Expression	[50] := 0.003		
C	Expression	[51] := 5		
D	Expression	[67] := 27		

The Branching statement aborts this DYO's execution when the Optimizer is running. When the Optimizer is running, it is the Optimizer that should assign its loop values to the variables. The variables would be assigned by the DYO when the Optimizer is not running.

Summary

Designing a profitable trading system can be rewarding and yet challenging. Many advanced users find the chase to be a lot of fun. Ensign has the tools which can be used to help you succeed in your research. As demonstrated in this example, DYOs can be used to implement the rules of a trading system. The program will accomplish the trades after stop and limit prices are set. And the DYO can execute Buy and Sell commands, and exit a position.

The trades that are made by the system post to a Trade Report which shows the details of the entry and exit prices, dates, times, and profits. The Optimizer is an additional power tool that can be used to research which parameters should be used in the system's design.

The Ensign team hopes you enjoy the power and flexibility available to you. Be creative and experience the exhilaration of researching and designing your own trading systems.

"Past Performance is not an indicator of future results."