

Adaptrade Builder

1.2

ADAPTRADE SOFTWARE

33175 Temecula Parkway #A, 336
Temecula, CA 92592

Internet: www.adaptrade.com

Phone: 951 302-1624

Fax: 208 977-1111

Email: info@adaptrade.com

Product: Strategic development software for writing EasyLanguage code for trading system development platform.

Requirements: Windows

Price: \$1,495 lifetime license

by Donald W. Pendergast Jr.

If you've been using TradeStation as a trading system development platform for a while and found yourself wondering how to expand your EasyLanguage (EL) skill sets in order to begin coding your own custom EL scripts, then you've probably considered using the services of an experienced EL programmer. While this can be a great way to have a trading system built for you in a short length of time, if the trading idea upon which the system is built is based on a flawed concept, then you've wasted your time and money in what amounts to a hit-or-miss approach to finding and profiting from a well-designed trading system.

However, there is another, less grueling pathway that aspiring developers can choose to travel, and that is Adaptrade's Builder, a standalone product for writing EasyLanguage. Builder assists with virtually every phase of crafting, testing, and then deploying trading models for TradeStation and MultiCharts, automatically writing the complete EL code for your desired strategy, no matter which market you may trade. In this review, we'll examine the key features of Builder and then take it a step farther by creating a viable trading strategy that can be tweaked by readers who wish to do additional research and development work.

OVERVIEW

Once you pay for and download the software from the Adaptrade website (there is also a free 14-day trial period), you'll have access to a full array of training resources and user guides, all of which are written in simple, clearly defined terms that even a budding system developer can grasp.

IMPORTING DATA

The first task in the build process is to select the market workspace and then import the data series for the stock or futures contract under examination. This is easily accomplished and will load right into Builder with a click of the Add button; the user then simply adjusts the settings for symbol, point value, and round-trip trading costs (slippage and commissions). Next, the user has the option to determine what the in-sample/out-of-sample proportion of the data series is to be, along with whether the out-sample period will occur at the beginning or the end of the data series.

STRATEGY OPTIONS

This is where the true innovation of Builder begins to shine. Users can define that only a certain mix of technical indicators and/or entry/exit order types be used to find the most appropriate system for a tradable in a certain time frame, or they can keep all of the default indicator and order types, trusting Builder to use its logic to home in on the trading strategy it deems most useful. Included are 23 technical indicators and nine order types, allowing for a myriad of potential system designs to be searched out. Users also have the ability to adjust the parameter

range setting on average true range (ATR) multiples and also on indicator and price pattern lookback periods.

Once the market data is imported and the mix of technical indicators and order types are defined by the user, the next step in the development process is to quantify the build metrics used to create trading systems with well-defined characteristics such as net profit, number of trades, profit factor, average trade size, drawdown, maximum adverse excursion, and other useful metrics.

After the build metrics are selected, the next step is to select the number of population and generation sizes for the genetic optimization. Additional options include the ability to adjust the tree and tournament sizes and the crossover/mutation percentages. Once all this is in place, the final step is to push the Build button and let the program work its wonders.

Depending on the size of the population/generation pool and the length of the dataset chosen, the system building process could last for a period of hours. Builder will keep you abreast as to the status of the process, and you'll be able to determine if you're going to need to let your computer run for an extended period of time, which depends on the complexity of the models you choose to develop for yourself and also on the speed and processing capabilities of your computer.

IN PROCESS

While Builder is running through all of the various genetic permutations allowed by your preselected build metrics, you'll be able to view each improvement in the



Builder assists with the crafting, testing, and then deploying trading models for TradeStation and MultiCharts.

equity curves that are generated as well as the performance of both in-sample and out-of-sample trades. These performance metrics are as follows:

- Net profit
- Number of trades
- Average trade
- Percent wins
- Profit factor
- Drawdown
- Correlation coefficient
- Significance
- Complexity
- Average win
- Average loss
- Win/loss ratio
- Return/drawdown ratio
- Average bars
- Average bars wins
- Average bars loss
- Max win
- Max loss
- Average MAE
- Fitness
- Max bars back

Serious system developers will already know which of these performance metrics matter most to their own unique systems. For novice developers, however, it's safe to say that they will mostly want to make sure the system is developed with a statistically significant number of trades across a variety of market conditions, and that they'll also want to keep a close eye on the drawdown, correlation coefficient, and significance factors as well, which all figure heavily in producing relatively straight, linear equity curves in Builder.

A REAL SYSTEM BUILD PROJECT

To demonstrate the potential effectiveness of Builder, I decided to develop a system for the Standard & Poor's 500 emini stock index futures market, running a generic build development process. Here are some of the key build metrics and parameters selected:

Ticker: @ES

Time frame: Five-minute bars

Trading window: 0705 to 1345 CT

Data used for testing: May 24, 2010,

FIGURE 1: INDICATOR AND ORDER BUILD SETS. With 23 technical indicators/tools and nine different ways to enter/exit trades, a nearly unlimited variety of trading concepts can be quickly put to the test in Builder.

FIGURE 2: BUILD GOALS. This particular build metric mix seeks to develop a system with high profitability, a large number of trades, a relatively smooth equity curve, and a minimum amount of inputs.

through May 23, 2011

Commissions and slippage, round-trip: \$25 per trade

Contract units: One

Initial capital: \$15,000

BUILD METRICS AND OPTIONS

Market sides: Long and short

Trading logic options: Long/short symmetry (same system logic for long and short entries/exits) and exit at end of the day

Parameter ranges: Minimum ATR multiples: 0.50. Maximum ATR multiples: 5.0. Maximum lookback, indicators: 100. Maximum lookback, price patterns: 20

Indicator build set: All 23 technical indicators enabled

Order build set: All nine entry/exit order types were enabled (Figure 1).

BUILD GOALS

In keeping with my own system design motto of "Keep it simple, safe, and profitable" (easier said than done!), only four of the 19 build metrics were enabled (Figure 2).

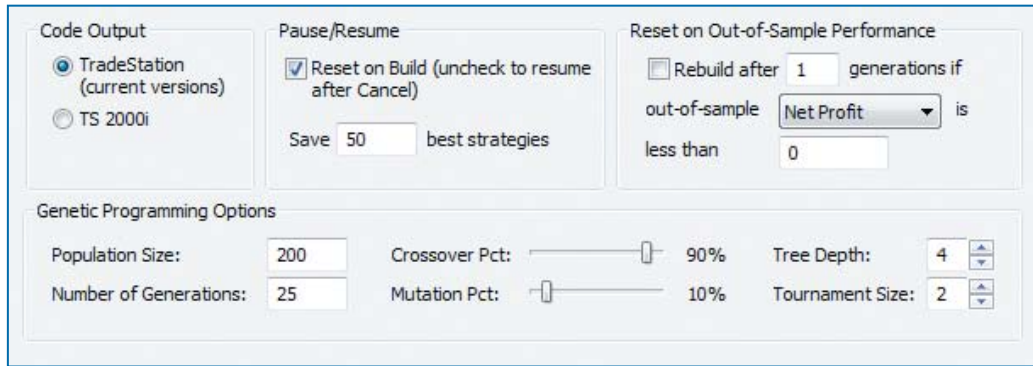
Net profit: Maximized and set to a weighting of 1.50

Number of trades: Maximized and set to a weighting of 2.00

Correlation coefficient: Maximized and set to a weighting of 2.50

Complexity: Minimized and set to a weighting of 1.50

PRODUCT REVIEW



Builder may be the perfect venue in which serious EasyLanguage developers can elevate their skills.



FIGURE 3: BUILD OPTIONS. Increasing the population and/or generation size may help Builder locate and develop more and better trading models, but this will also increase the time needed to do so by a proportionate amount.

BUILD OPTIONS

Genetic programming options:

Population size set to 200 and generations set to 25

Crossover percentage set to 90%

Mutation percentage set to 10%

Tree depth set to 4

Tournament depth set to 2

I also elected to save the 50 best strategies produced during Builder’s run (Figure 3). All of this seems overwhelming at first, but with the extensive quick start and user guides that come with Builder, most of the technical

concepts and development/testing procedures will be easy to pick up on within a few hours of experimentation with the program. The program also comes with a full set of example and bonus strategies that can help get new users comfortable with the vast array of features contained therein.

DOES IT WORK?

Let’s see what happened after I pushed the Build button and let Builder craft a viable emini S&P500 futures trading system for daytrading. The EasyLanguage code that the software developed for the strategy

can be found in sidebar “EasyLanguage Strategy Code For TradeStation.”

TIME SAVER

Very impressive! For a nonprogramming type, or for those not yet well-versed in EasyLanguage, Builder greatly simplifies what was once perhaps the most time-consuming aspect of system design, particularly in the case of highly complex systems with dozens or even hundreds of lines of code. From here, it’s a snap to simply copy the code from Builder and then paste it into your EasyLanguage document in TradeStation.

EASLANGUAGE STRATEGY CODE FOR TRADESTATION

```
{
EasyLanguage Strategy Code for TradeStation
Population member: 2

Created by: Adaptrade Builder version 1.1.1.0
Created: 5/24/2011 10:51:40 AM

TradeStation code for TS 6 or newer

Build Dates: 5/24/2010 to 5/23/2011
}

{ Strategy inputs }
Inputs: N1 (65),
        N2 (34),
        NBarEx (84);

{ Variables for entry conditions }
Var: EntCondL (false),
      EntCondS (false);

{ Entry conditions }
EntCondL = DMIMinus(N1) > DMIPlus(N2);

EntCondS = DMIMinus(N1) < DMIPlus(N2);
```

```
{ Entry orders }
If (MarketPosition = 0 or time = SessionEndTime(0, 1)) and EntCondL then
begin
    Buy next bar at market;
end;

If (MarketPosition = 0 or time = SessionEndTime(0, 1)) and EntCondS then
begin
    Sell short next bar at market;
end;

{ Exit orders, long trades }
If MarketPosition > 0 and time <> SessionEndTime(0, 1) then begin

    If BarsSinceEntry >= NBarEx then
        Sell next bar at market;
end;

{ Exit orders, short trades }
If MarketPosition < 0 and time <> SessionEndTime(0, 1) then begin

    If BarsSinceEntry >= NBarEx then
        Buy to cover next bar at market;
end;

SetExitOnClose;
```

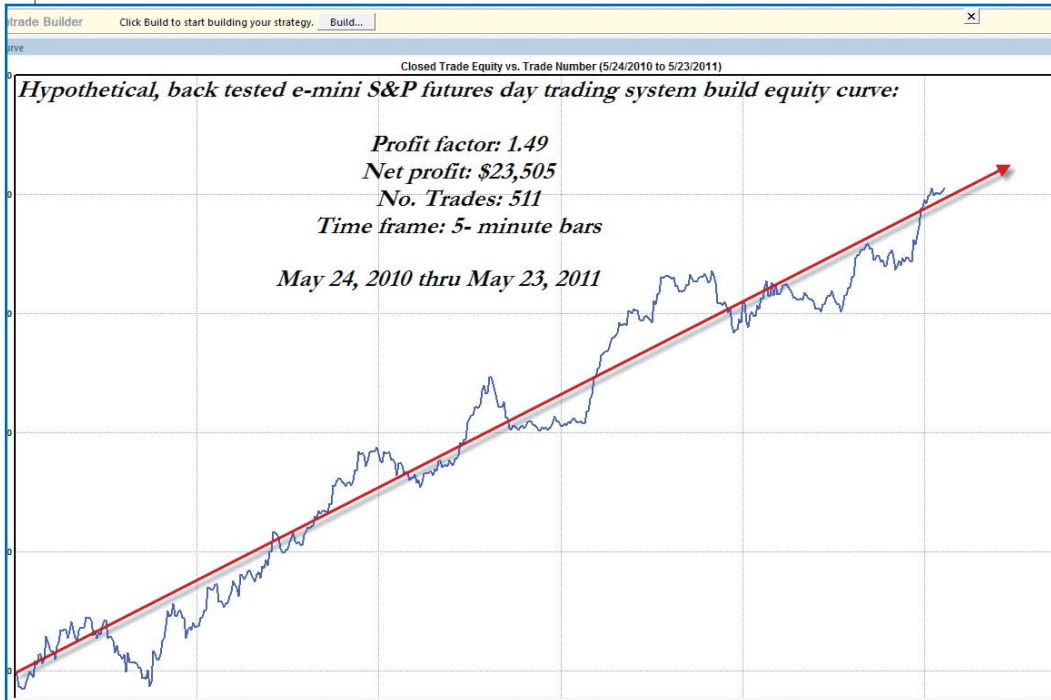


FIGURE 4: RESULTS. For such a simple system concept, the hypothetical, backtested equity curve is fairly straight, with reasonable drawdowns. Having more than 500 trades in the year-long back-run test helps increase the likelihood of statistical accuracy for the strategy.

RESULTS?

As you can see from Figure 4, the back-tested results are very encouraging, but remember that it is the forward-testing results (on new data that the system wasn't tested on) that matter most. As I write this in early June 2011, the system has generated an additional 10 trades since May 24, 2011, of which five were winners and five were losers. In pure forward-test mode since the system back-test was run, the strategy has returned \$937.50 in closed hypothetical profits during those 10 completed trades, before commissions and slippage.

No one knows if this rate of return will continue, of course, but the fact that the forward-test mode has met with success can only inspire confidence to those generating their own strategies with Builder. For the record, I added a \$500 stop-loss for the forward-testing phase; the backtest run was performed without using any fixed stop-loss exit.

SUMMARY

Adaptrade Builder is a well-rounded, feature-rich strategy development tool for serious TradeStation and MultiCharts users. At various times, I've used different EL programmers to code my own original trading concepts. On two occasions, the results were better than I had expected, but another time I learned that my basic system idea was flawed and not worth pursuing. However, the programmer still has to be paid (and the code they created for me worked flawlessly, by the way), whether or not the actual testing and trading results are successful. If you have Builder, however, once you own the program and can learn to use it successfully, your days of paying for EL programming are over — no more wasted development dollars. If the system idea doesn't work, just scrap it and start over again in Builder. Play with this software long enough and you might be amazed at some of the solid strategies it can produce in your favorite markets.

There is a learning curve involved, of course, but my experiences with Builder tell me that it's worth it to learn the program as you seek to create the best possible strategies for yourself and/or your clients. Support for the product is excellent, and all email and phone inquiries I made when seeking technical assistance were answered promptly, and the issues were cleared up quickly and easily.

POWER TO THE PEOPLE

Perhaps the significance of automatically writing EasyLanguage code so easily obtainable by the average systems developer/trader isn't such a

matter of import to the rest of the world, but for those who've struggled to make sense out of programming EasyLanguage code, it may be. And it may be just as important for those in the world of system trading who seek to make sense out of what can be an intimidating, time-consuming facet of trade system development.

Adaptrade's Builder software may be the perfect venue in which serious EasyLanguage developers can elevate their skills even as they learn new concepts that will likely cause them to view the field of trade system development in an entirely new light.

Donald W. Pendergast Jr. owns and operates Linear Trading Systems, based in Jacksonville, FL. He has several emini stock index futures and silver trading systems available for auto trading at Striker Securities in Chicago, IL. He may be reached at <http://linearjax.www.denovo7.com/> or at linearjax@gmail.com.

Find similar articles online at Traders.com

