

# AlgorithmicTrading.net

## Best-in-Class Trading Platform

**T**rading—regarded as one of the oldest professions in human history—has always been under scrutiny with unanswered questions, doubts, and insecurities. In particular, the futures market offers endless opportunities for tremendous profits, but benefits come with their own unique risk profiles. Often, there are debates among renowned industry pundits over the best methods to trade efficiently while minimizing risk. Tracing back history, an Italian mathematician, Leonardo Fibonacci once stated that many elements in the daily life of a person follow a predictable sequence. Similarly, traders and third-party trading system developers might exercise the same pattern as they plot sequential levels of support and resistance attempting to exploit reversals from previous trends. An electrical engineer by profession, Richard Metzger decided to step up the game and started a software company focused on providing best in class quantitative trading solutions by providing high quality algorithmic trading strategies. “Our desire is to enable our clients to generate alpha while doing our best to minimize the risk and emotions involved with trading,” says Metzger, the CEO of AlgorithmicTrading.net. Having worked as a logic design engineer for about 15 years, Metzger started writing and back-testing algorithms for use on the trade station platform and a company was born.

AlgorithmicTrading.net provides a 100-percent automated trading system which can be installed and loaded onto the trade station platform or automatically traded through one of the NFA registered brokers who support the trading algorithms. “If someone doesn’t utilize an algorithmic trading approach to the markets, they might still be doing very well, but the entire process is still an emotional roller coaster as profits are made and lost. With our product, the user knows that the algorithm runs repetitively without any human intervention, and this eventually removes one of the potential negative variables,” clarifies Metzger. As an example, The Swing Trader package utilizes his two best performing algorithms and provides a robust implementation for traders or investors, based purely on technical indicators and pattern

recognition logic that will wait for the most optimal setup before placing a trade. Also, users are able to track the daily progress of the quantitative trading algorithms using the OEC iBroker app as it sends push notifications onto the clients’ smart phone as new trades are placed. Customers also receive daily statements from the NFA registered clearing firm such as Gain capital or RJ Obrien. The solution itself is practically turnkey, allowing for a fast and easy account setup process guided by one of the NFA registered brokers who support the algorithms. In addition, the algorithms could be setup on the customers PC and auto-traded utilizing the trade station platform.



**Our desire is to enable our clients to generate alpha while doing our best to minimize the risk and emotions involved with trading**

While past performance is not indicative of future performance and customer experiences could vary from one customer to another, one of their first customers is still using the original NQ legacy algorithms and has had great success doing so.

Down the line, AlgorithmicTrading.net is looking to potentially hire a few additional engineers to help boost R&D efforts. Part of these new R&D efforts could include incorporating AI and machine learning algorithms. “AI and machine learning have some subtle disadvantages, but they also have the potential to reshape how quantitative trading is done in a positive way. Our intention is to provide the best possible algorithms we can develop, whether they are based on trades generated by a neural networks or from some other more traditional algorithmic trading style, our passion and goal is to provide value by delivering high quality algorithms which can beat the competition using the most objective standards.” concludes Metzger. **CM**



Richard Metzger