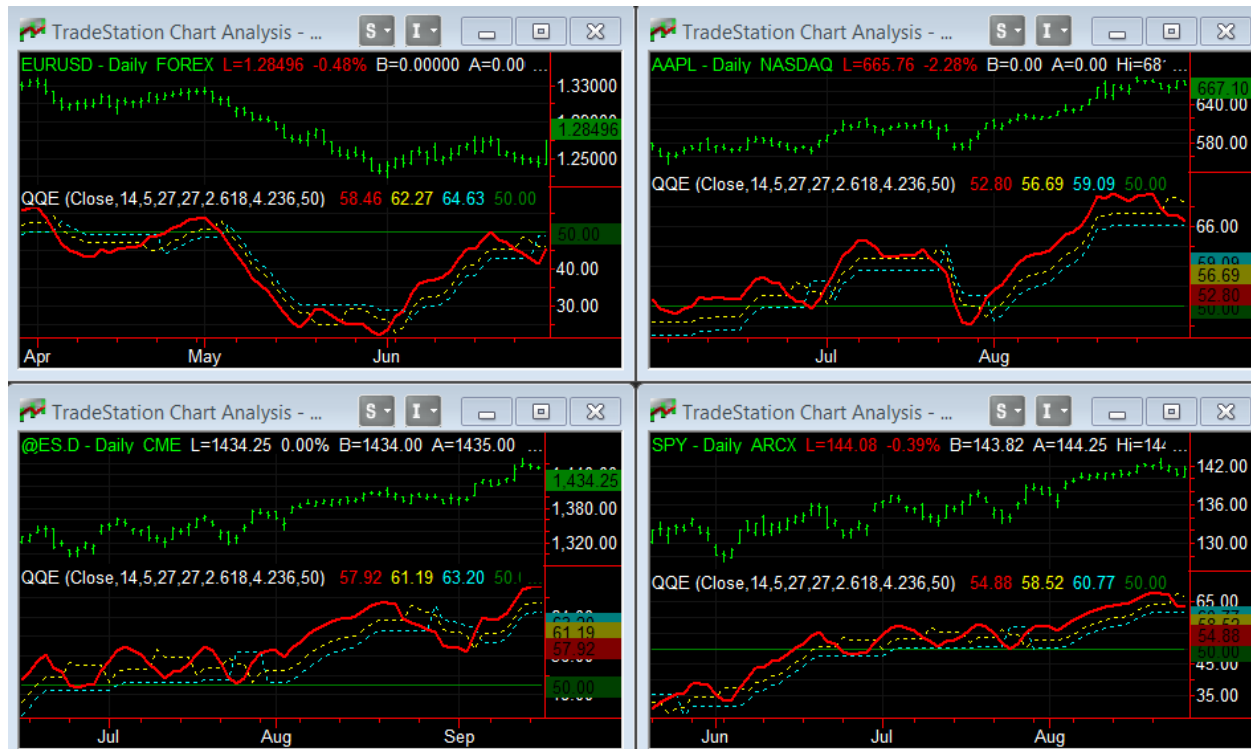


# Quantitative Qualitative Estimation (QQE) Indicator

Provided by John Bruch



TradeStation Requirements: TradeStation 9.1

**Important Disclosures:**

All support, education and training materials provided by TradeStation are for informational and educational purposes only, and are intended to help customers learn more about the TradeStation trading platform and TradeStation's brokerage services. No type of trading or investment advice is being given.

This material may also discuss in detail how TradeStation is designed to help you develop, test and implement trading strategies. However, TradeStation does not provide or suggest trading strategies. We offer you unique tools to help you design your own strategies and look at how they could have performed in the past. While we believe this is very valuable information, we caution you that simulated past performance of a trading strategy is no guarantee of its future performance or success. We also do not recommend or solicit the purchase or sale of any particular securities or derivative products. Any symbols referenced are used only for the purposes of the demonstration, as an example – not a recommendation.

Finally, this material may discuss automated electronic order placement and execution. Please note that even though TradeStation has been designed to automate your trading strategies and deliver timely order placement, routing and execution, these things, as well as access to the system itself, may at times be delayed or even fail due to market volatility, quote delays, system and software errors, Internet traffic, outages and other factors.

All proprietary technology in TradeStation is owned by our affiliate TradeStation Technologies, Inc. Equities, equities options and commodity futures products and services are offered by TradeStation Securities, Inc., a member of NYSE, FINRA, NFA and SIPC. Forex products and services are offered by TradeStation Forex, a division of IBFX, Inc. (member NFA). ©2012 TradeStation. All rights reserved.

# Quantitative Qualitative Estimation (QQE) Indicator

(EasyLanguage – 'QQE.eld') (Workspace – 'QQE.tsw')

Provided by John Bruch

The QQE is a bit of a mystery indicator based on Welles Wilder's RSI. While we have been unable to determine its origins, it seems to be something often discussed or requested on message boards and forums.

QQE is based on a smoothed RSI value and the smoothed RSI is plotted. Then, the absolute value of the change in smoothed RSI values from bar to bar is calculated and averaged twice. This double-smoothed value is then multiplied by two user-specified constants to create two additional lines that are also plotted (SlowTL and FastTL). Several inputs are provided so that the user can configure the indicator.

How the indicator is used varies from trader to trader; as a trending indication, look for the SmoothRSI to cross above or below the 50 level.

As an oscillator much like the RSI it is based on, you can watch for overbought or oversold conditions. Generally greater than or equal to 70 is overbought, while an oversold condition is generally less than or equal to 30.

Divergence is another way to use QQE; looking for divergences between the QQE momentum and price momentum can be effective in identifying potential reversal points.

Look for the SmoothRSI to cross above or below the SlowTL dotted plot for alert confirmations.

## Time Frames

Although not a lot has been written about QQE, it can be used on any symbol at any bar interval. Keep in mind that RSI (the basis for QQE) was developed using daily data, so that might be a good place to start here.



## Quantitative Qualitative Estimation (QQE) indicator

### Inputs

RSIPrice( Close )	{ the price to be used in the calculation of the RSI }
RSILength( 14 )	{ the number of bars to be used in the calculation of the RSI }
RSISmoothLength( 5 )	{ the length of the exponential moving average of the RSI; this value determines the smoothing factor used in the exponential moving average calculation }
ATRLength( 27 )	{ the length of the exponential moving average of the true range of the smoothed RSI }
ATRSmoothLength( 27 )	{ the length of the exponential moving average of the moving average of the true range of the smooth RSI }
FastATRMult( 2.618 )	{ a scaling factor that is multiplied by the moving average of the moving average of the true range of the smooth RSI }
SlowATRMult( 4.236 )	{ a scaling factor that is multiplied by the moving average of the moving average of the true range of the smooth RSI }
SmoothRSIAlertLevel( 50 )	{ if alerts are enabled for the indicator, an alert will be triggered if the smoothed RSI crosses the level specified by this input }

### Plots

Plot1( SmoothRSI )  
Plot2( FastTL )  
Plot3( SlowTL )  
Plot4( SmoothRSIAlertLevel)

### Alert Criteria

if SmoothRSI crosses over SmoothRSIAlertLevel  
if SmoothRSI crosses under SmoothRSIAlertLevel

## Function Provided

The QQE calculations are provided as an EasyLanguage function so that you can use this idea in strategies or other indicators.

## Function EL Code

```
inputs:
double RSIPrice( Close ), { the price to be used in the calculation of the RSI }
int RSILength( 14 ), { the number of bars to be used in the calculation of the RSI }
int RSISmoothLength( 5 ), { the length of the exponential moving average of the
    RSI; this value determines the smoothing factor used in the exponential moving
    average calculation }
int ATRLength( 27 ), { the length of the exponential moving average of the
    true range of the smoothed RSI }
int ATRSmoothLength( 27 ), { the length of the exponential moving average of the
    moving average of the true range of the smooth RSI }
double FastATRMult( 2.618 ), { a scaling factor that is multiplied by the moving
    average of the moving average of the true range of the smooth RSI }
double SlowATRMult( 4.236 ), { a scaling factor that is multiplied by the moving
    average of the moving average of the true range of the smooth RSI }
double SmoothRSIAlertLevel( 50 ) ; { if alerts are enabled for the indicator, an
    alert will be triggered if the smoothed RSI crosses the level specified by this
    input }

variables:
    double RetVal( 0 ), double oSmoothRSI( 0 ),
    double oFastTL( 50 ),double oSlowTL( 50 ),
    intrabarpersist bool OkToPlot( false ) ;

RetVal = QQE( RSIPrice, RSILength, RSISmoothLength, ATRLength, ATRSmoothLength,
    FastATRMult, SlowATRMult, oSmoothRSI, oFastTL, oSlowTL ) ;

{ do not plot until all exponential moving averages have stabilized }

once( CurrentBar > 4 * MaxList( RSILength, RSISmoothLength, ATRLength,
    ATRSmoothLength ) )
    OkToPlot = true ;

if OkToPlot then begin

    Plot1( oSmoothRSI, "SmoothRSI" ) ;
    Plot2( oFastTL, "FastTL" ) ;
    Plot3( oSlowTL, "SlowTL" ) ;
    Plot4( SmoothRSIAlertLevel, "SmRSIAlrtLev" ) ;

{ alerts }
if AlertEnabled then
    if oSmoothRSI crosses over SmoothRSIAlertLevel then
        Alert( "SmoothRSI crossing over " + NumToStr(SmoothRSIAlertLevel, 5 ) )
    else if oSmoothRSI crosses under SmoothRSIAlertLevel then
        Alert( "SmoothRSI crossing under " + NumToStr(SmoothRSIAlertLevel, 5 ) );

end ; // if OkToPlot
```