

<b>Instructions</b>	
Product:	OptionsX
Description:	OptionsX is a toolset for automated options trading relative to underlying symbols in TradeStation Charts and RadarScreen. It is designed for traders looking to programmatically enhance the buying power and profit potential of their systems through options.
Date:	October 15, 2019
Contact Information:	<a href="#">!QCL Solutions, LLC</a>
Document Notes:	<i>(Tips &amp; Tricks are italicized and in parentheses)</i>
Disclaimer:	This software should not be construed as providing trading or investment recommendations. Any decisions you make using this software are done solely at your own risk. The risk of loss in trading can be substantial. Furthermore, traders may sustain losses greater than their investments, regardless of which asset classes are traded. In particular, trading options is not suitable for all. Please click here to view the document entitled <a href="#">Characteristics and Risks of Standardized Options</a> . Before trading, carefully consider inherent risks in light of your financial condition.

### Prerequisite Installation

1. TradeStation 9.1 (Update 29) or later required.
2. TradeStation real-time data subscriptions for symbols to be traded.

### **Overview**

OptionsX is a toolset for automated options trading relative to underlying symbols in TradeStation Charts and RadarScreen. Among the tools are EL functions that can be used in Strategies and Indicator. The primary functions are opChain, opChainData, opData, opDataHiDayVol, qcTrader and qcTraderEX. The first four are for pulling data for options symbols and automatically locating options symbols that meet your criteria. The last two are for trading; they keep track of your orders and position information even through platform restarts. opStrategiesMonitor is for automatically converting Strategy signals to Option trades. Specify expiration dates out and strike levels up or down, for it to locate appropriate options and trade within your pricing rules. And opMPTrader is also a plug-n-play trading system, operating as an indicator that can be used in Charts and RadarScreen. Just plug in your strategy functions, like the example opMPT\_RSI\_Signal function. The idea of OptionsX is robust trading through simplicity, with access to details if needed. With OptionsX you can easily leverage options to diversify, control risk, reduce margin exposure, and multiply returns (or shorten the time necessary) to achieve what is possible with your strategies.

### **The Toolbox**

To help you get started, contact us at [Support@QCLsolutions.com](mailto:Support@QCLsolutions.com). Also visit [OptionsX at QCLsolutions.com](#) for EL examples. And if you want additional tools for the Toolbox, we are ready to support. Here are the current tools:

Name	Description
opChainEX TA_OptionsX_opChainEX	A TradingApp that displays option chain symbols, based on function inputs, by strike price and expiration date.
opChainDemo TA_OptionsX_opChainDemo	An indicator demonstrating the opChain and opChainData functions for accessing entire option chains and detailed data for each available option.
opChainDemoEX TA_OptionsX_opChainDemoEX	An indicator demonstrating the opChain and opChainDataEX functions for accessing entire option chains and detailed data (including TradeStation quote field data for Implied Volatility) for each available option, along with custom calculated values for Black-Scholes price, the Greeks and Implied Volatility.
opDemo TA_OptionsX_opDemo	An indicator demonstrating the opData function for locating options and collecting data.
opSymbolDemo TA_OptionsX_opSymbolDemo	An indicator demonstrating the opSymbolData function for pulling option symbol data along with that of its underlying symbol.
opSymbolDemoEX TA_OptionsX_opSymbolDemoEX	An indicator demonstrating the opSymbolDataEX function for pulling option symbol data along with that of its underlying symbol, and options complex calculations.
qcRank TA_OptionsX_qcRank	An indicator utilizing the opRank function. It was intended for demonstration but is practical for regular use. Put your custom value function in the MyValueToRank input.
opStrategiesMonitor TA_OptionsX_opStrategiesMonitor	A plug-n-play trading system that can turn any TradeStation Strategy into an Options trading system, automatically converting equity Strategy signals into Option trades. Specify expiration dates out and strike levels up or down. Then once it has positions and orders, it keeps track of them, even through computer restarts. (Smart inputs: iAccount = "" auto-detects default account from the underlying, SpreadMin/Max as minimum moves of the underlying, Chain_BasisPrice < 0 to automatically set as Last of underlying, and Chain_BasisELDate < 730101 EL date to specify as a CurrentDate offset.)

opMPTTrader TA_OptionsX_opMPTTrader	A plug-n-play trading indicator that can automatically trade options for an underlying symbol based on input Call and Put market-position functions.
opMPT_RSI_Signals TA_OptionsX_opMPT_RSI_Signals	An indicator for displaying values of the opMPT_RSI_Signal function.
opMPT_RSI_Signals TA_OptionsX_opMPT_RSI_Signals	A strategy using the opMPT_RSI_Signal function for trading the underlying like opMPTTrader would trade options relative to the underlying. It can be used to optimize opMPT_RSI_Signal inputs for use with opMPTTrader.
opMPT_RSI_Signal TA_OptionsX_opMPT_RSI_Signal	An example market-position (i.e. signal) function for using with opMPTTrader.
qcRank TA_OptionsX_qcRank	A global ranking function for sharing information between multiple applications, enabling each to know how their value compares to others by absolute and percentile rank. It may be used to compare trade signals in different applications to decide which ones to use.
qcRankEX TA_OptionsX_qcRankEX	A global ranking function for sharing information between multiple applications, enabling each to know how their value compares to others by absolute and percentile rank. It may be used to compare trade signals in different applications to decide which ones to use. Returns include a vector of values from various contributing applications, along with a provider for vector updated events.
qcAlertBar TA_OptionsX_qcAlertBar	Sends Alerts, issuing according to inputs for uniqueness and frequency.
qcAlertOK TA_OptionsX_qcAlertOK	Uses the same logic as qcAlertBar, issuing according to inputs for uniqueness and frequency. This function can accumulate an Alert string in cases where there may be multiple alerts from the same code cycle.
qcAlertOKsend TA_OptionsX_qcAlertOKsend	Using logic like qcAlertBar and qcAlertOK, this function can send, as a TradeStation Alert, the accumulated string of qcAlertOK. Then clears the string afterward for further use. As opposed to qcAlertOK, this function uses the AlertText input only if there is an accumulated string passing logic criteria. It

	then appends any input AlertText to the accumulated string, to be issued in a TradeStation Alert.
opChain TA_OptionsX_opChain	A function for pulling an entire option chain of data, made available to opChainData functions. (Used in opChainDemo and opChainDemoEX indicators.)
opChainData TA_OptionsX_opChainData	A function for accessing option data made available from an opChain function. With extended quotes as in EX functions described below. (Used in opChainDemo indicator.)
opChainDataEX TA_OptionsX_opChainDataEX	A function for accessing option data made available from an opChain function like opChainData, with the additional returns of the TradeStation quote field data for Implied Volatility, along with custom calculated values for Black–Scholes price, the Greeks and Implied Volatility. (Used in opChainDemoEX indicator.)
opChainEX TA_OptionsX_opChainEX	A function for pulling an entire option chain of data returned with symbol attributes, a dictionary of chain symbols records and list of available expiration dates. (Used with the opChainEXsymbol function to pull symbols.)
opChainEXSymbol TA_OptionsX_opChainEXSymbol	A function to pull option symbols from the dictionary records returned by opChainEX.
opSymbolData TA_OptionsX_opSymbolData	A function for pulling option symbol data along with that of its underlying symbol.
opSymbolDataEX TA_OptionsX_opSymbolDataEX	A function for pulling option symbol data along with that of its underlying symbol, and options complex calculations.
opSymbolRoot TA_OptionsX_opSymbolRoot	A function to determine an option symbol root from the underlying symbol.
opClickTip TA_OptionsX_opClickTip	A tool for creating popup windows with a click where your application is applied. It displays when clicking on any plot or on a specific plot number or name. (As for strategies, it can display on any chart click with first input of "ClickAnywhere".)
opClickRtnDouble TA_OptionsX_opClickRtnDouble	A tool for creating popup windows with a click where your application is applied. It displays when clicking on any plot or on a specific plot number or name. (As for strategies, it can display on any chart click with first input of "ClickAnywhere".) The function returns a double value from a supplied list, matching your menu item list,

	when a menu item is clicked.
opData TA_OptionsX_opData	An options symbol locator and data collector. FOR STOCK OPTIONS ONLY. (Used in opDemo indicator.)
opDataHiDayVol TA_OptionsX_opDataHiDayVol	An options symbol locator and data collector, locating the nearest options contract with higher daily volume than surrounding strike prices. FOR STOCK OPTIONS ONLY. (Used in opMPTrader.)
opDataEX TA_OptionsX_opDataEX	An options symbol locator and data collector with extended quotes including Option Symbol, Category, Strike, ExpirationDate, BigPointValue, Last, LastTradeVol, LastTradeDT, Ask, AskSize, AskDT, Bid, BidSize, BidDT, OpenInt, DailyOpen, DailyHigh, DailyLow, and DailyVolume. FOR STOCK OPTIONS ONLY.
opDataHiDayVolEX TA_OptionsX_opDataHiDayVolEX	An options symbol locator and data collector, locating the nearest options contract with higher daily volume than surrounding strike prices. With extended quotes including Option Symbol, Category, Strike, ExpirationDate, BigPointValue, Last, LastTradeVol, LastTradeDT, Ask, AskSize, AskDT, Bid, BidSize, BidDT, OpenInt, DailyOpen, DailyHigh, DailyLow, and DailyVolume. FOR STOCK OPTIONS ONLY.
qcCALmanager TA_OptionsX_qcCALmanager	A function for managing the number of simultaneous data provider loading operations which can be an issue with multiple instances in RadarScreen.
qcCALmanagerC TA_OptionsX_qcCALmanagerC	A function, equivalent to qcCALmanager using slightly different mechanism that tend to be more efficient in certain scenarios.
qcDateNTimeOfDT TA_OptionsX_qcDateNTimeOfDT	Returns ELDate and ELTime from a double value like ComputerDateTime.
qcDateTimeOfEL TA_OptionsX_qcDateTimeOfEL	Returns a double DateTime value from ELDate and ELTime.
qcDateTimeOfStr TA_OptionsX_qcDateTimeOfStr	Returns a double DateTime value from a string representation of Date and Time.
qcDateTimeToStr TA_OptionsX_qcDateTimeToStr	Returns a string value for a number, with the appropriate decimal precision.

qcDateTimeTS TA_OptionsX_qcDateTimeTS	Returns computer system time, adjusted to best match TradeStation time stamps.
qcDecimals TA_OptionsX_qcDecimals	Returns the decimal precision of the current symbols price data.
qcDecimalsEX TA_OptionsX_qcDecimalsEX	Returns the decimal precision of an input number, up to an input maximum.
qcELTimeDiff TA_OptionsX_qcELTimeDiff	Returns the difference between two EL-format time inputs in seconds, minutes or hours.
qcELTimeInTimeRange TA_OptionsX_qcELTimeInTimeRange	Returns true if the current bar date and time stamp is within a specified time range relative the current day and active trading session. (Similar to qcNowInTimeRange.)
qcGetAccount TA_OptionsX_qcGetAccount	Returns the active or default account using logic based on where the function is being used.
qcGetAccountOfCategory TA_OptionsX_qcGetAccountOfCategory	Returns the default account for an input symbol category or security type. If invalid the symbol category or security type of the analysis technique symbol is used.
qcGetAccountOfChart TA_OptionsX_qcGetAccountOfChart	Returns the active chart account.
qcGetAccountOfSymbol TA_OptionsX_qcGetAccountOfSymbol	Returns the default account for an input symbol, along with additional symbol information such as the trading symbol, security type and decimal precision.
qcGetAccountOfSymbolEX TA_OptionsX_qcGetAccountOfSymbolEX	Returns the default account for an input symbol, along with additional symbol information such as the trading symbol, security type and decimal precision. Also provides with updated events.
qcAccountBySymbol TA_OptionsX_qcAccountBySymbol	Returns the default account for an input symbol, without additional symbol information, for ease of use such as in an application input.
qcGetSetRecord TA_OptionsX_qcGetSetRecord	A utility for storing and retrieving variable information, that can maintain persistence of systems through platform shutdowns, restarts and even crashes. Can also be used by other windows setup to use stored records.

qcGetSetRecords TA_OptionsX_qcGetSetRecords	A utility for storing and retrieving variable information like qcGetSetRecord. This version can store and retrieve a range of settings. It is used by qcGetSetRecord.
qcGetSetRecordsEX TA_OptionsX_qcGetSetRecordsEX	An advanced version of qcGetSetRecords, for dynamically managing multiple simultaneous read/write operations to/from storage.
qcGetSetRecordsEXff TA_OptionsX_qcGetSetRecordsEXff	An advanced version of qcGetSetRecords, for dynamically managing multiple simultaneous read/write operations to/from storage, utilizing a fast-file read/write system.
qcGetSetRecordsBase TA_OptionsX_qcGetSetRecordsBase	A utility which is a base version of qcGetSetRecords. It can be used for additional control of, and access to, settings file locations and names.
qcGetSetRecordsBaseEX TA_OptionsX_qcGetSetRecordsBaseEX	A utility which is a base version of qcGetSetRecordsEX. It can be used for additional control of, and access to, settings file locations and names.
qcInstance TA_OptionsX_qcInstance	A utility for tracking the instance of itself within an analysis technique. If used multiple times in an indicator, each instance will have a different number to distinguish it.
qcInstanceEX TA_OptionsX_qcInstanceEX	A utility for tracking the instance of itself within an analysis technique. If used multiple times in an indicator, each instance will have a different number to distinguish it. This extended version provides additional information in reference inputs.
qcInstanceChecker TA_OptionsX_qcInstanceChecker	A utility function for instance control of an EL application or items within an application, where only one instance should be active at a time. It uses unique identifiers of applications and even items of applications for instance control, alerting in case of duplicates.
qcInstanceGUID TA_OptionsX_qcInstanceGUID	A utility for assigning and managing globally unique identifiers to the application in which it runs and to the instance of itself within the application (since it may be called in multiple places). It returns additional identifying information as is used by qcInstance, qcInstanceEX and other items.
qclsNew TA_OptionsX_qclsNew	A user-convenience function for detecting if the function's input value is new, relative to its specific instance by application and instance within that application (which could

	be as an input to the application). The identification of a new input is maintained through data refreshing events and even platform restarts.
qclsNewOnce TA_OptionsX_qclsNewOnce	A user-convenience function for detecting if the function's input value is new, relative to its specific instance by application and instance within that application (which could be as an input to the application). The identification of a new input is maintained through data refreshing events and even platform restarts. If a new input is identified, the True return is held until the function is uninitialized.
qcListNums TA_OptionsX_qcListNums	A utility function for tracking, managing and pulling information from a custom list of numbers.
qcNowAhead TA_OptionsX_qcNowAhead	Returns the difference between the computer system clock and TS time, as a double value in days. Also returns the difference in seconds and gives the latest system clock time of measurement.
qcNowInTimeRange TA_OptionsX_qcNowInTimeRange	Returns true if the current time is within a specified time range relative the current day and active trading session.
qcNowTimeDiffOfDT TA_OptionsX_qcNowTimeDiffOfDT	Returns a difference value in seconds, minutes or hours between the current time and an input double DateTime value.
qcNowToSessionClose TA_OptionsX_qcNowToSessionClose	Returns true if the current time is within an active trading session, at or after a specified BeginTime and at or before session close. Session beginning and ending times can be auto-detected (with -1 inputs) or overridden.
qcNumToStr TA_OptionsX_qcNumToStr	A convenience function for converting numbers to string values for display. It automatically trims to the decimal significance of the basis symbol, or trims simply trims off unnecessary 0's.
qcNumToStrMinDecm TA_OptionsX_qcNumToStrMinDecm	A convenience function for converting numbers to string values for display. It automatically trims off unnecessary 0's beyond the MinDecimals specified.
opOptionsComplex TA_OptionsX_opOptionsComplex	Returns true for valid inputs, providing custom calculated values for Black–Scholes price, the Greeks and Implied Volatility.
opQuotes TA_OptionsX_opQuotes	Returns quotes for an input symbol, including Option Symbol, Category, Strike,



	ExpirationDate, BigPointValue, Last, Ask, Bid, OpenInt, DailyOpen, and DailyVolume.
opQuotesEX TA_OptionsX_opQuotesEX	Returns quotes for an input symbol, including Option Symbol, Category, Strike, ExpirationDate, BigPointValue, Last, LastTradeVol, LastTradeDT, Ask, AskSize, AskDT, Bid, BidSize, BidDT, OpenInt, DailyOpen, DailyHigh, DailyLow, and DailyVolume.
qcQuoteLast TA_OptionsX_qcQuoteLast	Returns the quote value of Last for an input symbol, along with event information.
qcLastBySymbol TA_OptionsX_qcLastBySymbol	Returns the Last quote for an input symbol, without additional information, for ease of use such as in an application input.
opSADVolatility TA_OptionsX_opSADVolatility	A function, primarily for internal use, that returns the statistical annualized daily volatility for an underlying symbol as is commonly used in options pricing models.
opSNprefix TA_OptionsX_opSNprefix	The TradingApp Store function, strategy or indicator prefix
opSNSuffix TA_OptionsX_opSNSuffix	The TradingApp Store function, strategy or indicator name suffix
OptionsX TA_OptionsX_OptionsX	The TradingApp Store product name
qcFundamentalQuoteDoubleWU TA_OptionsX_qcFundamentalQuoteDoubleWU	A function, primarily for internal use, that returns double-value quote data for an input fundamental quote field.
qcPositionBase TA_OptionsX_qcPositionBase	Returns Positon records for an input symbol and account (or the default account of the symbol if not specified).
qcPositionBaseEX TA_OptionsX_qcPositionBaseEX	Returns Positon records for an input symbol and account (or the default account of the symbol if not specified). Returns include the PositionsProvider, so that a calling application can handle events.
qcPositionChange TA_OptionsX_qcPositionChange	A strategy monitoring function which returns true for any position change, and provides the position quantity with appropriate sign.
qcPositionChangeEX TA_OptionsX_qcPositionChangeEX	A strategy monitoring function which returns true for any position change. It also provides the position quantity with appropriate sign along with last trade price, date & time, and quantities (two in the case of U-turns).
qcPositionChangeInStrategy TA_OptionsX_qcPositionChangeInStrategy	A strategy monitoring function like qcPositionChange, but only for use within Strategy code where it can also access

	simulated trades in which there was no net change in position or net profit.
qcPositionChangeInStrategyEX TA_OptionsX_qcPositionChangeInStrategyEX	A strategy monitoring function like qcPositionChangeEX, but only for use within Strategy code where it can also access simulated trades in which there was no net change in position or net profit.
qcSAP_MinMovePts TA_OptionsX_qcSAP_MinMovePts	A utility to determine the minmove points and related properties of a symbol.
qcSLView TA_OptionsX_qcSLView	A utility function of opChain and related functions.
qcStringClean TA_OptionsX_qcStringClean	A convenience function that may help clients with the RootOptionSymbol input of opChain, opChainData, opChainDataEX, related functions and applications.
qcSymbolAttributesProvider TA_OptionsX_qcSymbolAttributesProvider	A utility to load and notify when a SymbolAttributesProvider is loaded.
qcTBillRateNow TA_OptionsX_qcTBillRateNow	A function, primarily for internal use, that uses web services to locate the latest T-Bill coupon equivalent Rate for the 4, 13 (default), 26 or 52-week maturity. These rates, esp. for the 13-week maturity, are commonly used in options pricing models.
qcTextToScreen TA_OptionsX_qcTextToScreen	A convenience function for displaying text on a chart; left, right or center of locations specified as 0 to 1 horizontally and vertically on the current view.
qcTrader TA_OptionsX_qcTrader	An automated trading function, which manages and reports positions it creates. It also reports account position. It further files 'owned' position records for continuity between recalculation and platform restarts.
qcTraderEX TA_OptionsX_qcTraderEX	An automated trading function like qcTrader, with extended ownership controls to protect against potentially conflicting OptionsX trading systems.
qcTradeBot TA_OptionsX_qcTradeBot	The underlying function for qcTrader and qcTraderEX, also used directly by opStrategiesMonitor.
qcLastBarOnChartS TA_OptionsX_qcLastBarOnChartS	Like LastBarOnChart, a built-in EasyLanguage function.
qcVolatility TA_OptionsX_qcVolatility	A function, primarily for internal use, that returns statistical volatility for a symbol, interval, session, look-back period, data field, etc. Also provides the raw data mean.

Utilizing the above tools is **opStrategiesMonitor**, for you to easily trade options based on a strategy applied to the underlying symbol. opStrategiesMonitor is a companion strategy, applied one or more times in the same chart as your strategies, that can locate, trade and track option positions relative to underlying trade signals. Inputs are as follows:

Inputs	Demo Values	Notes
_____OPTIONSX_____	"_____"	An inputs dialog label only.
OptionsTradingOK_if1	0	After setting Strategy Properties for All... Enable order placement objects. Then can set this input to 1 to enable trading.
iAccount	""	The account number for trading. "" auto-detects the default account for the UnderlyingSymbol.
CustomID	""{To ignore positions from prior days, include string elements like NumToStr(CurrentDate,0)}	A custom identifier to further distinguish a StrategiesMonitor instance for storing and persisting position information. It is particularly useful when using multiple instances in a chart, such as in creating option spread positions.
TradeSignalTypes	0	Controls whether All, Long or Short underlying strategy signals are to be used for option trading. 0 for "All", "1" or more for "Long", -1 or less for "Short".
TradeHistoricSignal	0	A switch to control how historical strategy signals are handled. 0 = real-time signals only. 1 = If OpenPositionProfit >= TradeHistcOPosProfit (one time) -1 = If OpenPositionProfit <= TradeHistcOPosProfit (one time) 2 = If OpenPositionProfit >= TradeHistcOPosProfit (keep looking until there is a realtime signal) -2 = If OpenPositionProfit <= TradeHistcOPosProfit (keep looking until there is a realtime signal) 3 = If OpenPositionProfit >= TradeHistcOPosProfit or OpenPositionProfit <= -TradeHistcOPosProfit (one time) -3 = If OpenPositionProfit <= TradeHistcOPosProfit and OpenPositionProfit >= -TradeHistcOPosProfit (one time) 4 = If OpenPositionProfit >= TradeHistcOPosProfit or OpenPositionProfit <= -TradeHistcOPosProfit (keep looking until there is a realtime signal) -4 = If OpenPositionProfit <= TradeHistcOPosProfit and OpenPositionProfit >= -TradeHistcOPosProfit (keep looking until there is a realtime signal) +/- 5 or more for trading on any historical signal when reach realtime.

TradeHistcOPosProfit	0	Works with TradeHistoricSignal as noted above.
OverridePosRcd_onChg	0	A switch to modify records of owned positions for the StrategiesMonitor instance. And integer change in the value signal to the instance, one time only, that the Override_Owned_Qty in the input below is to be the new owned position quantity of the StrategiesMonitor instance.
Override_Owned_Qty	0	Works with OverridePosRcd_onChg as noted above.
SpreadXpct_BuyToOPEN	50	For buy-to-open orders: The degree of aggressiveness in executing orders with limit prices, relative to the inside bid and ask at time of placement. 0 specifies limit at bid, 100 for limit at ask, -100 for bid-(ask-bid), 200 for ask+(ask-bid), etc. (Set to >=999999.99 for market orders)
SpreadXpct_SellToClose	150	For sell-to-close orders: The degree of aggressiveness in executing orders with limit prices, relative to the inside bid and ask at time of placement. 0 specifies limit at ask, 100 for limit at bid, -100 for ask+(ask-bid), 200 for bid-(ask-bid), etc. (Set to >=999999.99 for market orders)
SpreadXpct_SellToOPEN	50	For sell-to-open orders: The degree of aggressiveness in executing orders with limit prices, relative to the inside bid and ask at time of placement. 0 specifies limit at ask, 100 for limit at bid, -100 for ask+(ask-bid), 200 for bid-(ask-bid), etc. (Set to >=999999.99 for market orders)
SpreadXpct_BuyToClose	150	For buy-to-close orders: The degree of aggressiveness in executing orders with limit prices, relative to the inside bid and ask at time of placement. 0 specifies limit at bid, 100 for limit at ask, -100 for bid-(ask-bid), 200 for ask+(ask-bid), etc. (Set to >=999999.99 for market orders)
SpreadXpct_EODtoClose	500	For SetExitOnClose orders: The degree of aggressiveness in executing orders with limit prices, relative to the inside bid and ask at time of placement. 0 specifies not crossing the bid-ask spread (negative values can also be used, for orders away from market). 100 enables limit orders to cross the spread (like "Hit/Take" orders), and greater than 100 allows for crossing of the spread plus (as with "Hit/Take+" orders). (Set to >=999999.99 for market orders)
SpreadMinimumMM	1	Works with the above SpreadXpct_ inputs for a minimum (ask-bid) spread used in calculations. Expressed as a number of underlying minimum moves.

SpreadMaximumMM	100	Works with the above SpreadXpct_ inputs for a maximum (ask-bid) spread used in calculations. Expressed as a number of underlying minimum moves.
UnderlyingSymbol	Symbol	The underlying symbol to be used for locating option contracts. Symbol is an EL reserved word that locates the symbol of the chart where applied. Other valid symbol may be used, being input between double quotes, like "SPY".
Chain_CustomRootCall	""	A custom root name for call option contracts which may differ from that of the underlying symbol.
Chain_CustomRootPut	""	A custom root name for put option contracts which may differ from that of the underlying symbol.
Chain_CustomRootExt	""	A custom root name extension for option contracts which is not part of the underlying symbol.
Chain_BasisPrice	-1{Last}	A basis price for relative referencing of option contracts with the Option_StrikeLevelsUp input below. The Option_StrikeLevelsUp input can be positive or negative to locate contracts a number of strike levels up or down relative to the Chain_BasisPrice. Note that a negative value triggers auto-detection of Last quotes for the underlying symbol.
Chain_PriceToStrikeM	1	Works with the above Chain_BasisPrice input to allow location of strike prices when may be a multiple of the basis price. The input is normally 1.
Chain_BP_PrcntAbove	25	Works with the above Chain_BasisPrice and Chain_PriceToStrikeM inputs to set the upper range of strike prices to consider in pulling option chain data. Expressed as a percentage of the effective basis price.
Chain_BP_PrcntBelow	25	Works with the above Chain_BasisPrice and Chain_PriceToStrikeM inputs to set the lower range of strike prices to consider in pulling option chain data. Expressed as a percentage of the effective basis price.

Chain_BasisELDate	2	A basis date for relative referencing of option contracts with the Option_ExpDatesOut input below. The Option_ExpDatesOut input is generally positive, but can also be negative, to locate contracts a number of expiration dates forward or backward relative to the Chain_BasisELDate. Note that inputs less than 730101 (for January 1st of the year CBOE began listing), including negative inputs, trigger auto-detection of the date as an input number of days offset relative to the CurrentDate, i.e. CalcDate(CurrentDate, Chain_BasisELDate).
Chain_BD_DaysForward	31	Works with the effective Chain_BasisELDate from the input above, to set the number of days forward to consider for expiration dates in pulling option chain data. Regardless of the basis date, chain data is populated with the CurrentDate forward to the effective Chain_BasisELDate plus Chain_BD_DaysForward.
Chain_ContractDays	"3456"	For specifying contract expiration days to consider in pulling option chain data. "" includes all days, like "0123456". 0 for Sunday through 6 for Saturday. "3456" for Wednesday through Saturday expiration days. Weekly, monthly and quarterly options can have distinctive expiration days. Weekly usually expire on Wednesday or Fridays. Monthly options usually expire of Thursday or Saturday. Quarterly options expire on the last Friday of the quarter.
Chain_UseFileSystem	FALSE	Set to TRUE for XML file storage of key option chain data to speed future lookups. But since option chain details can change from day to day, and since option chain lookups have been well optimized, it is generally recommended that the FALSE setting be used.
Option_FindCall	TRUE	TRUE to trade Call options, and FALSE to trade Put options.
Option_InvertSig4Put	TRUE	TRUE to invert strategy Buy signals to SellShort and Sell signals to BuyToCover, when trading Put options. FALSE not reverse signals and trade put options as directly indicated by strategy signals and the Option_QtyMultiple below.

Option_QtyMultiple	1	With stocks, a 100 share underlying position translates to 1 option contract. That 1 contract can be multiplied for leveraged trading. It can also be used for inverted trading with negative values, similar to the above setting for puts (where TRUE is the same as a -1 value here for put contracts only).
Option_StrikeLevelsUp	0	The number of strike levels up (+/-), relative to the effective Chain_BasisPrice, to locate for trading.
Option_ExpDatesOut	0	The number of expiration dates out (+/-), relative to the effective Chain_BasisELDate, to locate for trading.
StatusLoc_Xright	0	For locating the status line relative to the upper-left corner of the chart, expressed a number of pixels to the right of the left edge.
StatusLoc_Ydown	20	For locating the status line relative to the upper-left corner of the chart, expressed a number of pixels below the top. It can depend on font size, but with setting below, each status line generally fits well with 20 pixels of room for each.
StatusAlign_Horiz	1	For locating the status line. 1 specifies above the reference point, 0 centered on the reference point, -1 below the reference point.
StatusAlign_Vert	1	For locating the status line. 1 specifies to the right of the reference point, 0 centered on the reference point, -1 to the left of the reference point.
StatusColor_ON	"Yellow"	For status line text color when trading is enabled.
StatusColor_OFF	"Magenta"	For status line text color when trading is disabled.
StatusFont	"Arial"	For status line text font type.
StatusFont_Size	10	For status line text font size.
ChangeToRecalculate	0	Change this input to force an indicator re-initialization without reloading data.

## Demonstration EL

Below is an EL example of using opData to locate and display options data:

---

```
{ opDemo for instructions }
```

Inputs:

```
    UnderlyingSymbol(Symbol),  
    FindCall(True),  
    BeginSearchELDate(CalcDate(CurrentDate, 1)),  
    EndSearchELDate(CalcDate(CurrentDate, 93)),  
    ContractDays("3456"),  
    BeginSearchStrike(Last),  
    EndSearchStrike(Last + 25),  
    IncSearchStrikeCstm(0);
```

Variables:

```
    Ready(False),  
    FindNew(False),  
    OT_Symbol(""),  
    OT_Category(254),  
    OT_Strike(0),  
    OT_ExpirationELDate(0),  
    OT_BigPointValue(0),  
    OT_Last(0),  
    OT_Ask(0),  
    OT_Bid(0),  
    OT_OpenInt(0),  
    OT_DailyVolume(0),  
    OT_DailyOpen(0),  
    OT_FoundDT(0),  
    OT_Message(""),  
    vBeginSearchStrike(0),  
    vEndSearchStrike(0),  
    vIncSearchStrikeCstm(0);
```

```
// Locate option and get data updates afterward
```

```
Once Begin
```

```
    vBeginSearchStrike = BeginSearchStrike;  
    vEndSearchStrike = EndSearchStrike;  
    vIncSearchStrikeCstm = IncSearchStrikeCstm;
```

```
End;
```

```
Ready = opData (    FindNew,  
                   OT_Symbol,  
                   OT_Category,  
                   OT_Strike,
```



```

        OT_ExpirationELDate,
        OT_BigPointValue,
        OT_Last,
        OT_Ask,
        OT_Bid,
        OT_DailyOpen,
        OT_DailyVolume,
        OT_OpenInt,
        OT_FoundDT,
        OT_Message,
        UnderlyingSymbol,
        FindCall,
        BeginSearchELDate,
        EndSearchELDate,
        ContractDays,
        vBeginSearchStrike,
        vEndSearchStrike,
        vIncSearchStrikeCstm, 0);

// Plot
If
    OT_Symbol <> ""
Then Begin
    Plot1(OT_Symbol, "Symbol");
    Plot2(qcNumToStr(OT_Category, False), "Category");
    Plot3(qcNumToStr(OT_Strike, True), "Strike");
    Plot4(qcNumToStr(OT_ExpirationELDate, False), "ExpELDate");
    Plot5(qcNumToStr(OT_BigPointValue, False), "BPV");
    Plot6(OT_Last, "Last");
    Plot7(OT_Ask, "Ask");
    Plot8(OT_Bid, "Bid");
    Plot9(OT_DailyOpen, "DailyOpen");
    Plot10(qcNumToStr(OT_DailyVolume, False), "DayVolume");
    Plot11(qcNumToStr(OT_OpenInt, False), "OpenInt");
    Value0 = Maxlist(0, (ComputerDateTime - OT_FoundDT) * 1440);
    Plot12(NumToStr(Value0, 2), "MinsAgo");
    Condition0 = opClickTip(
        ""{or can use "ClickAnywhere"}, OT_Symbol+","-,"
        +qcNumToStr(OT_Ask, True)+ " ask,-,"
        +qcNumToStr(OT_Last, True)+ " last,-,"
        +qcNumToStr(OT_Bid, True)+ " bid", "");
End;

```

---

Below is an EL example of using opChain and opChainData to pull an option chain and display data:

---

```
{ opChainDemo for instructions }
```

Inputs:

```
// Pull Options Chains
string UnderlyingSymbol(Symbol),
string CallRootSymbolSpec(""),
string PutRootSymbolSpec(""),
string SymbolSpecExtra(""),
double BasisPrice(Last),
double PriceToStrikeMultplr(1),
double BP_PercentAbove(25),
double BP_PercentBelow(25),
int BasisELDate(CurrentDate),
int BD_DaysForward(15),
string ContractDays("3456"),
// Get Data from Chain
bool FindCall(True),
int StrikeLevelsUp(0),
int ExpDatesOut(0),
bool UseFileSystem(False),
// ClickTip
bool ShowClickTip(False);
```

Variables:

```
string sCLICKANYWHERE("CLICKANYWHERE"),
//opChain
intrabarpersist bool Ready(False),
intrabarpersist bool FindNew(False),
intrabarpersist int BeginSearchELDate(0),
intrabarpersist int EndSearchELDate(0),
intrabarpersist double BeginSearchStrike(0),
intrabarpersist double EndSearchStrike(0),
tsdata.marketdata.SymbolAttributes ocUnderSymbolAttr(Null),
intrabarpersist bool ocRecordsAvailable(False),
intrabarpersist double ocFoundDT(0),
intrabarpersist string ocMessage(""),
tsdata.common.DataProvider ocUpdated(Null),
//opChainData
intrabarpersist bool ReadyData(False),
tsdata.marketdata.SymbolAttributes oUnderSymbolAttr(Null),
intrabarpersist string oOptionSymbol(""),
```

```

intrabarpersist int oOptionCategory(0),
intrabarpersist double oStrike(0),
intrabarpersist int oExpirationDate(0),
intrabarpersist double oBigPointValue(0),
intrabarpersist double oLast(0),
intrabarpersist int oLastTradeVol(0),
intrabarpersist double oLastTradeDT(0),
intrabarpersist double oAsk(0),
intrabarpersist int oAskSize(0),
intrabarpersist double oAskDT(0),
intrabarpersist double oBid(0),
intrabarpersist int oBidSize(0),
intrabarpersist double oBidDT(0),
intrabarpersist int64 oOpenInt(0),
intrabarpersist double oDailyOpen(0),
intrabarpersist double oDailyHigh(0),
intrabarpersist double oDailyLow(0),
intrabarpersist int64 oDailyVolume(0),
intrabarpersist double oUpdatedDateTime(0),
intrabarpersist int oMaxStrikeLevelsUp(0),
intrabarpersist int oMaxStrikeLevelsDown(0),
intrabarpersist int oMaxExpDatesForward(0),
intrabarpersist int oMaxExpDatesBack(0),
intrabarpersist string oMessage(""),
tsdata.common.DataProvider oUpdated(Null);

// Pull option chain and get data once available
If 0 < BasisPrice then Begin
    Once Begin
        BeginSearchStrike = BasisPrice * MaxList(1 - BP_PercentBelow
            / 100, 0);
        EndSearchStrike = BasisPrice * MaxList(1 + BP_PercentAbove
            / 100, 0);
        BeginSearchELDate = BasisELDate;
        EndSearchELDate = CalcDate(BasisELDate,
            MaxList(BD_DaysForward, 0));
    End;
    Ready = opChain(FindNew,
        UnderlyingSymbol,
        CallRootSymbolSpec,
        PutRootSymbolSpec,
        SymbolSpecExtra,
        iff(FindCall, 1, -1),
        PriceToStrikeMultplr,

```

```

BeginSearchELDate,
EndSearchELDate,
ContractDays,
BeginSearchStrike,
EndSearchStrike,
UseFileSystem,
ocUnderSymbolAttr,
ocRecordsAvailable,
ocFoundDT,
ocMessage,
ocUpdated);

If Ready then Begin
  ReadyData =
    opChainData(UnderlyingSymbol,
      CallRootSymbolSpec,
      PutRootSymbolSpec,
      SymbolSpecExtra,
      FindCall,
      BasisPrice,
      PriceToStrikeMultplr,
      StrikeLevelsUp,
      BasisELDate,
      ExpDatesOut,
      ContractDays,
      oUnderSymbolAttr,
      oOptionSymbol,
      oOptionCategory,
      oStrike,
      oExpirationDate,
      oBigPointValue,
      oLast,
      oLastTradeVol,
      oLastTradeDT,
      oAsk,
      oAskSize,
      oAskDT,
      oBid,
      oBidSize,
      oBidDT,
      oOpenInt,
      oDailyOpen,
      oDailyHigh,
      oDailyLow,
      oDailyVolume,

```

```

        oMaxStrikeLevelsUp,
        oMaxStrikeLevelsDown,
        oMaxExpDatesForward,
        oMaxExpDatesBack,
        oUpdatedDateTime,
        oMessage,
        oUpdated);

    End;
End;

// Plot
Plot1(elsystem.DateTime.Now.ToString(), "Now");
If Ready then
    Plot2("Ready", "ChainReady")
Else
    Plot2("", "ChainReady");
If
    0 < ocFoundDT
Then Begin
    Value0 = Maxlist(0, (ComputerDateTime - ocFoundDT) * 1440);
    Plot11(NumToStr(Value0, 2), "ChainMinsAgo");
    Plot12(ocMessage, "ChainMessage");
    If ReadyData then Begin
        // Symbol Properties
        Plot22(oOptionSymbol, "Symbol");
        Plot23(NumToStr(oOptionCategory, 0), "Category");
        Plot24(qcNumToStr(oStrike, True)+" ["
            + NumToStr(StrikeLevelsUp,0)+"]", "Strike");
        Plot25(NumToStr(oExpirationDate,0)+" ["
            + NumToStr(ExpDatesOut,0)+"]", "ExpELDate");
        Plot26(qcNumToStr(oBigPointValue, False), "BPV");
        // Last
        Plot31(oLast, "Last");
        Plot32(NumToStr(oLastTradeVol, 0), "LastVol");
        Plot33(qcDateTimeToStr(oLastTradeDT), "LastDT");
        // Ask
        Plot41(oAsk, "Ask");
        Plot42(NumToStr(oAskSize, 0), "AskSize");
        Plot43(qcDateTimeToStr(oAskDT), "AskDT");
        // Bid
        Plot51(oBid, "Bid");
        Plot52(NumToStr(oBidSize, 0), "BidSize");
        Plot53(qcDateTimeToStr(oBidDT), "BidDT");
        // Additional Details

```

```

Plot61(oDailyOpen, "DailyOpen");
Plot62(oDailyHigh, "DailyHigh");
Plot63(oDailyLow, "DailyLow");
Plot64(NumToStr(oDailyVolume, 0), "DayVolume");
Plot65(NumToStr(oOpenInt, 0), "OpenInt");
Plot66(NumToStr(oMaxStrikeLevelsUp, 0), "MaxStrikesUp");
Plot67(NumToStr(oMaxStrikeLevelsDown, 0), "MaxStrikesDN");
Plot68(NumToStr(oMaxExpDatesForward, 0), "MaxExpDtsFwd");
Plot69(NumToStr(oMaxExpDatesBack, 0), "MaxExpDtsBCK");
If ShowClickTip then
    Condition0 = opClickTip(sCLICKANYWHERE,
                            oOptionSymbol
                            + ",-",
                            +qcNumToStr(oAsk,True)+" ("
                            + oAskSize.ToString()
                            + ") ask,-,"
                            +qcNumToStr(oLast,True)+" ("
                            + oLastTradeVol.ToString()
                            + ") last,-,"
                            +qcNumToStr(oBid,True)+" ("
                            + oBidSize.ToString()
                            + ") bid", "");
    // Reset sCLICKANYWHERE on plotted bar close
    sCLICKANYWHERE = "";
End;
If 0 < oUpdatedDateTime then Begin
    Value0 = Maxlist(0, (ComputerDateTime - oUpdatedDateTime)
                    * 86400);
    Plot71(NumToStr(Value0, 0), "SecSinceUpdt");
    Plot72(oMessage, "DataMessage");
End;
End;

```

---

Below is an EL example of using opChain and opChainDataEX to pull an option chain and display data, including the Greeks, Implied Volatility and associated values:

---

```

{ opChainDemoEX for instructions }
Inputs:
    // Pull Options Chains
    string UnderlyingSymbol(Symbol),
    string CallRootSymbolSpec(""),
    string PutRootSymbolSpec(""),

```

```

string SymbolSpecExtra(""),
double BasisPrice(Last),
double PriceToStrikeMultplr(1),
double BP_PercentAbove(25),
double BP_PercentBelow(25),
int BasisELDate(CurrentDate),
int BD_DaysForward(15),
string ContractDays("3456"),
// Get Data from Chain
bool FindCall(True),
int StrikeLevelsUp(0),
int ExpDatesOut(0),
double Rate100(0),
bool UseFileSystem(False),
// ClickTip
bool ShowClickTip(False);

```

Variables:

```

string sCLICKANYWHERE("CLICKANYWHERE"),
//opChain
intrabarpersist bool Ready(False),
intrabarpersist bool FindNew(False),
intrabarpersist int BeginSearchELDate(0),
intrabarpersist int EndSearchELDate(0),
intrabarpersist double BeginSearchStrike(0),
intrabarpersist double EndSearchStrike(0),
tsdata.marketdata.SymbolAttributes ocUnderSymbolAttr(null),
intrabarpersist bool ocRecordsAvailable(False),
intrabarpersist double ocFoundDT(0),
intrabarpersist string ocMessage(""),
tsdata.common.DataProvider ocUpdated(Null),
//opChainDataEX
intrabarpersist bool ReadyData(False),
tsdata.marketdata.SymbolAttributes oUnderSymbolAttr(null),
intrabarpersist string oOptionSymbol(""),
intrabarpersist int oOptionCategory(0),
intrabarpersist double oStrike(0),
intrabarpersist int oExpirationDate(0),
intrabarpersist double oBigPointValue(0),
intrabarpersist double oLast(0),
intrabarpersist int oLastTradeVol(0),
intrabarpersist double oLastTradeDT(0),
intrabarpersist double oAsk(0),
intrabarpersist int oAskSize(0),

```

```

intrabarpersist double oAskDT(0),
intrabarpersist double oBid(0),
intrabarpersist int oBidSize(0),
intrabarpersist double oBidDT(0),
intrabarpersist int64 oOpenInt(0),
intrabarpersist double oImpliedVoltyTS(0),
intrabarpersist double oDailyOpen(0),
intrabarpersist double oDailyHigh(0),
intrabarpersist double oDailyLow(0),
intrabarpersist int64 oDailyVolume(0),
intrabarpersist int oMaxStrikeLevelsUp(0),
intrabarpersist int oMaxStrikeLevelsDown(0),
intrabarpersist int oMaxExpDatesForward(0),
intrabarpersist int oMaxExpDatesBack(0),
intrabarpersist bool getComplex(True),
intrabarpersist string setAssetType(""),
intrabarpersist double setYield100(0),
intrabarpersist double setForeignRate100(0),
intrabarpersist double setVolty100(0),
intrabarpersist bool setIsEuropean(False),
intrabarpersist double oDaysLeft(0),
intrabarpersist double oOpPriceCalculated(0),
intrabarpersist double oDelta(0),
intrabarpersist double oGamma(0),
intrabarpersist double oVega(0),
intrabarpersist double oTheta(0),
intrabarpersist double oRho(0),
intrabarpersist bool getImpliedVolatility(True),
intrabarpersist double oImpliedVolatility(0),
intrabarpersist bool getProbabilityITMony(True),
intrabarpersist double oProbabilityITMoney(0),
intrabarpersist double oUpdatedDateTime(0),
intrabarpersist string oMessage(""),
tsdata.common.DataProvider oUpdated(Null);

// Pull option chain and get data once available
If 0 < BasisPrice then Begin
    Once Begin
        BeginSearchStrike = BasisPrice * MaxList(1 - BP_PercentBelow
            / 100, 0);
        EndSearchStrike = BasisPrice * MaxList(1 + BP_PercentAbove
            / 100, 0);
        BeginSearchELDate = BasisELDate;
        EndSearchELDate = CalcDate(BasisELDate,

```



```

        MaxList(BD_DaysForward, 0));
End;
Ready = opChain(FindNew,
                UnderlyingSymbol,
                CallRootSymbolSpec,
                PutRootSymbolSpec,
                SymbolSpecExtra,
                iff(FindCall, 1, -1),
                PriceToStrikeMultplr,
                BeginSearchELDate,
                EndSearchELDate,
                ContractDays,
                BeginSearchStrike,
                EndSearchStrike,
                UseFileSystem,
                ocUnderSymbolAttr,
                ocRecordsAvailable,
                ocFoundDT,
                ocMessage,
                ocUpdated);
If Ready then Begin
    ReadyData =
        opChainDataEX(UnderlyingSymbol,
                      CallRootSymbolSpec,
                      PutRootSymbolSpec,
                      SymbolSpecExtra,
                      FindCall,
                      BasisPrice,
                      PriceToStrikeMultplr,
                      StrikeLevelsUp,
                      BasisELDate,
                      ExpDatesOut,
                      ContractDays,
                      oUnderSymbolAttr,
                      oOptionSymbol,
                      oOptionCategory,
                      oStrike,
                      oExpirationDate,
                      oBigPointValue,
                      oLast,
                      oLastTradeVol,
                      oLastTradeDT,
                      oAsk,
                      oAskSize,

```

```

        oAskDT,
        oBid,
        oBidSize,
        oBidDT,
        oOpenInt,
        oImpliedVoltyTS,
        oDailyOpen,
        oDailyHigh,
        oDailyLow,
        oDailyVolume,
        oMaxStrikeLevelsUp,
        oMaxStrikeLevelsDown,
        oMaxExpDatesForward,
        oMaxExpDatesBack,
        getComplex,
        setAssetType,
        Rate100,
        setYield100,
        setForeignRate100,
        setVolty100,
        setIsEuropean,
        oDaysLeft,
        oOpPriceCalculated,
        oDelta,
        oGamma,
        oVega,
        oTheta,
        oRho,
        getImpliedVolatility,
        oImpliedVolatility,
        getProbabilityITMony,
        oProbabilityITMoney,
        oUpdatedDateTime,
        oMessage,
        oUpdated,
        oUpdated);
    End;
End;

// Plot
Plot1(elsystem.DateTime.Now.ToString(), "Now");
If Ready then
    Plot2("Ready", "ChainReady")
Else

```

```

    Plot2("", "ChainReady");
If
    0 < ocFoundDT
Then Begin
    Value0 = Maxlist(0, (ComputerDateTime - ocFoundDT) * 1440);
    Plot11(NumToStr(Value0, 2), "ChainMinsAgo");
    Plot12(ocMessage, "ChainMessage");
    If ReadyData then Begin
        // Symbol Properties
        Plot22(oOptionSymbol, "Symbol");
        Plot23(NumToStr(oOptionCategory, 0), "Category");
        Plot24(qcNumToStr(oStrike, True)+" ["
            + NumToStr(StrikeLevelsUp,0)+"]", "Strike");
        Plot25(NumToStr(oExpirationDate,0)+" ["
            + NumToStr(ExpDatesOut,0)+"]", "ExpELDate");
        Plot26(qcNumToStr(oBigPointValue, False), "BPV");
        // Last
        Plot31(oLast, "Last");
        Plot32(NumToStr(oLastTradeVol, 0), "LastVol");
        Plot33(qcDateTimeToStr(oLastTradeDT), "LastDT");
        // Ask
        Plot41(oAsk, "Ask");
        Plot42(NumToStr(oAskSize, 0), "AskSize");
        Plot43(qcDateTimeToStr(oAskDT), "AskDT");
        // Bid
        Plot51(oBid, "Bid");
        Plot52(NumToStr(oBidSize, 0), "BidSize");
        Plot53(qcDateTimeToStr(oBidDT), "BidDT");
        // Additional Details
        Plot61(oDailyOpen, "DailyOpen");
        Plot62(oDailyHigh, "DailyHigh");
        Plot63(oDailyLow, "DailyLow");
        Plot64(NumToStr(oDailyVolume, 0), "DayVolume");
        Plot65(NumToStr(oOpenInt, 0), "OpenInt");
        Plot66(NumToStr(oMaxStrikeLevelsUp, 0), "MaxStrikesUp");
        Plot67(NumToStr(oMaxStrikeLevelsDown, 0), "MaxStrikesDN");
        Plot68(NumToStr(oMaxExpDatesForward, 0), "MaxExpDtsFwd");
        Plot69(NumToStr(oMaxExpDatesBack, 0), "MaxExpDtsBCK");
        If ShowClickTip then
            Condition0 = opClickTip(sCLICKANYWHERE,
                oOptionSymbol
                + ",-"
                +qcNumToStr(oAsk,True)+" ("
                + oAskSize.ToString()

```

```

        + ") ask,-,"
+qcNumToStr(oLast,True)+" ("
        + oLastTradeVol.ToString()
        + ") last,-,"
+qcNumToStr(oBid,True)+" ("
        + oBidSize.ToString()
        + ") bid","");
// Reset sCLICKANYWHERE on plotted bar close
sCLICKANYWHERE = "";
End;
If 0 < oOpPriceCalculated then Begin
// The Greeks and Implied Volatility
Plot71(oOpPriceCalculated, "OpPriceCalcd");
Plot72(oDelta, "Delta");
Plot73(oGamma, "Gamma");
Plot74(oVega, "Vega");
Plot75(oTheta, "Theta");
Plot76(oRho, "Rho");
Plot77(oImpliedVolatility, "ImpVolty");
Plot78(oProbabilityITMoney, "ProbiltyInTM");
Plot79(oDaysLeft, "DaysLeft");
End;
If 0 < oUpdatedDateTime then Begin
Value0 = Maxlist(0, (ComputerDateTime - oUpdatedDateTime)
* 86400);
Plot91(NumToStr(Value0, 0), "SecSinceUpdt");
Plot92(oMessage, "DataMessage");
End;
End;

```

---

Note that if you copy-paste the above into TradeStation, that "" may be pasted as ". To resolve this in the TradeStation Development Environment, Edit → Replace... → Replace All:

- 1) (" with ("")
- 2) (" with ("",
- 3) ," with ,"")

For additional examples of using opChain, opChainData, opChainDataEX, opData, opDataHiDayVol, opStrategiesMonitor, qcTrader, qcTraderEX, opMPTrader and other OptionsX tools, contact us at: [Support@QCLsolutions.com](mailto:Support@QCLsolutions.com)  
And visit us at: [OptionsX at QCLsolutions.com](http://OptionsX at QCLsolutions.com)