

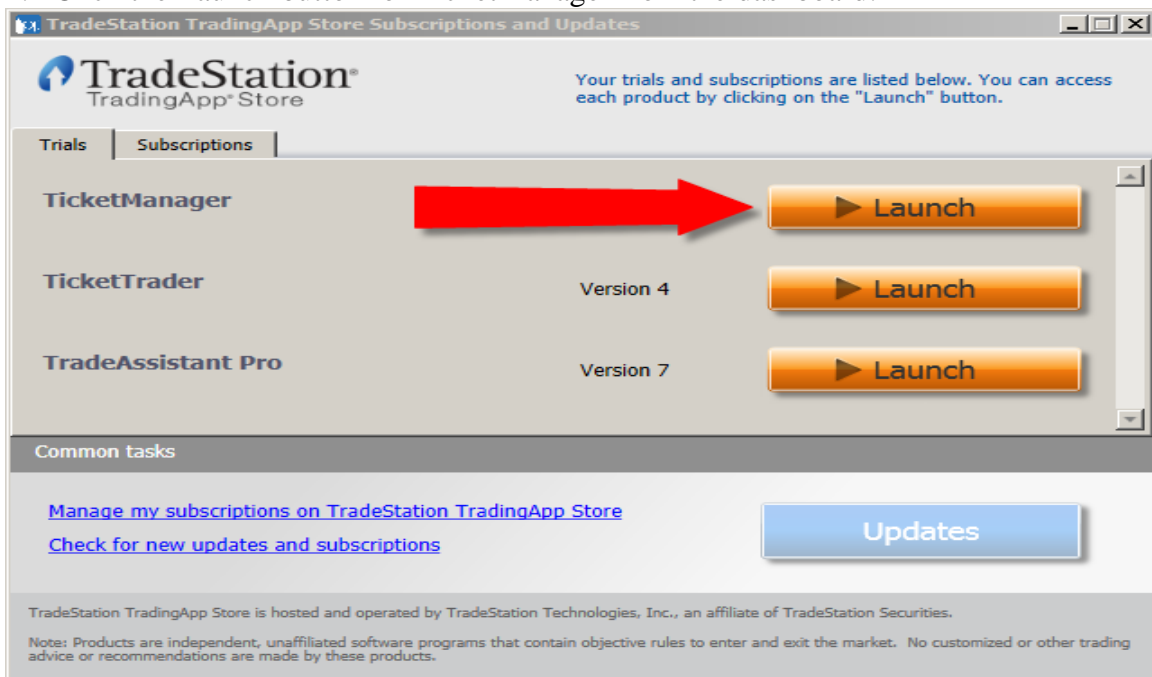
Instructions	
Product:	TicketManager
Date:	March 30, 2018
Purpose:	Professional Algorithmic Auto-Trading of TradeStation Strategies for robust scalable operations with minimal market impact. TicketManager aggregates multiple strategies on multiple markets and symbols. A low-maintenance solution that traders can trust to monitor various data sources and manage myriad trading details.
Contact:	<a href="#">!QCL Solutions, LLC</a>
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. Leveraged trading such as through options, futures, and forex exposes traders to heightened risk. Furthermore, traders may sustain losses greater than their investments, regardless of which asset classes are traded. Before trading, carefully consider inherent risks given your financial condition.

### Prerequisite Installation

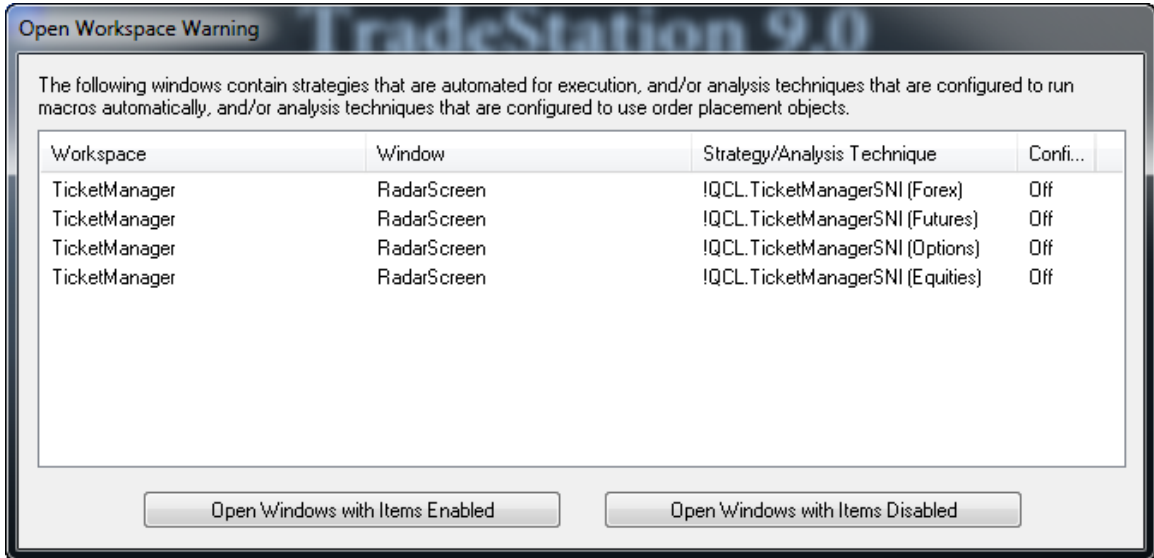
1. TradeStation 9.1 (Update 29) or later required.
2. A RadarScreen subscription. TicketManager is available to run in Charts but is intended for best operation in RadarScreen.
3. TradeStation real-time data subscriptions for symbols to be traded.

### Launching from a TradingApp Store prompt

1. **For initial testing, log into your Trading Simulation accounts.**
2. Click the Launch button for TicketManager from the dashboard:



3. A prompt like the one below will be raised, click “Open Windows with Items Enabled”:

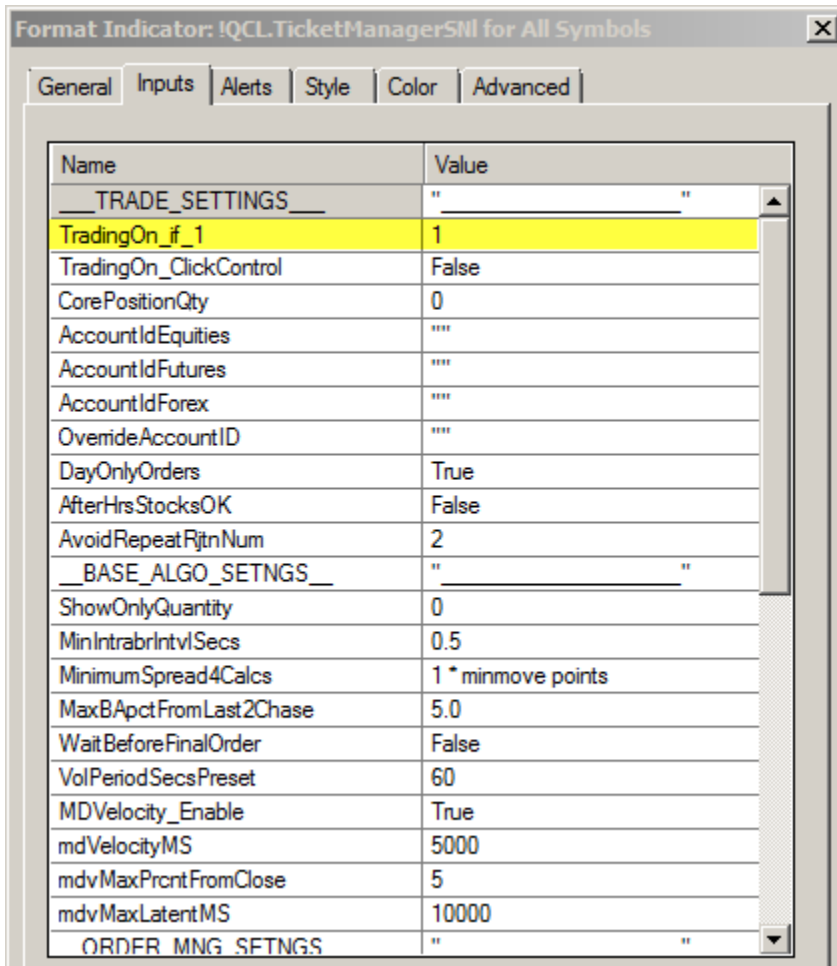


4. The workspace opened contains the following demonstration items:



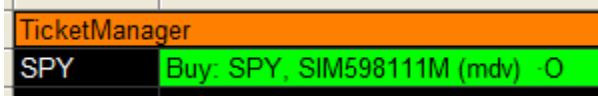
5. The default workspace has order objects enabled, as approved in step 3, but still needs to be enabled for algorithmic trading of Ticket instructions from strategies. To enable TicketManager to trade, click in the first plot column of TicketManager in the RadarScreen page. Then follow the menu sequence: Format → Analysis Technique → '!QCL.TicketManagerSNI' for 'SPY' (or for All Symbols...)

- When the Format Indicator dialog is raised, select the Inputs tab. Select the value for the first input and set to 1: TradingOn\_if\_1 = 1



- Then click OK. During an active equity market session, within a few minutes the demonstration workspace should produce Ticket instructions. Open a TradeManager window to view trades placed in the simulation account.
- Note that when the Communication Owning instance of StrategiesMonitor is sending Ticket instructions to TicketManager, in the upper-right margin of the chart you should see the following: **\*\*\* TicketManager: Receiving, ACTIVE \*\*\***

9. And in TicketManager you should see sell or buy instructions highlighted as here:



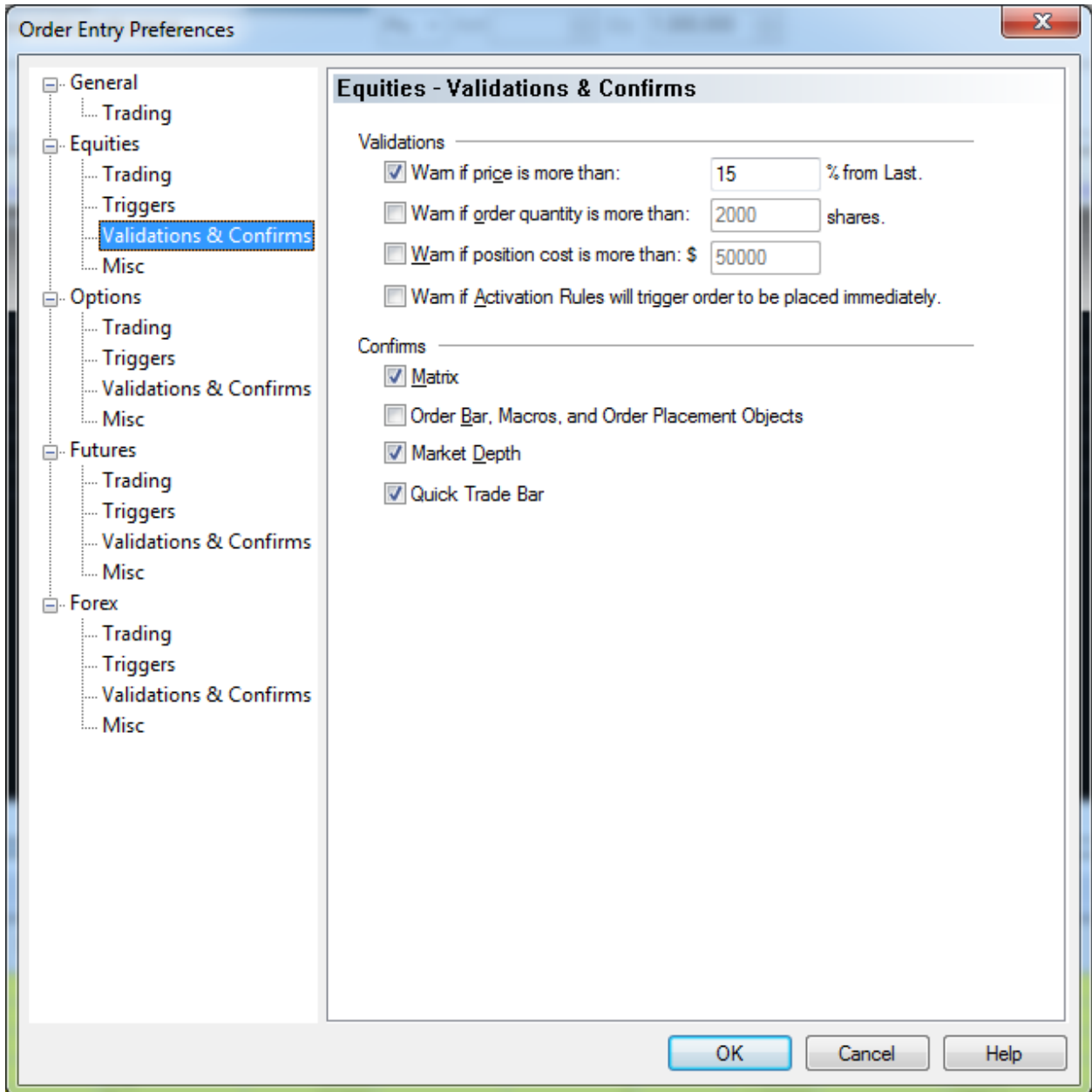
With details about the Ticket in the numerous RadarScreen plot columns of TicketManager and in the lower-right margin of the Communication Owning chart as here:



10. Note that a demonstration strategy is already applied to the two charts in the demonstration workspace. These strategies can be replaced as you might prefer. *(Do not enable Strategy Automation as this can disable the Communication Owning instance of StrategiesMonitor from sending Ticket instructions, on the possibility that TradeStation Strategy Automation could generate orders that are duplicated by TicketManager.)*
11. Open a TradeManager window to observe orders issued and managed by the TicketManager solution.
12. Note that StrategiesMonitor will aggregate strategy trades from multiple charts with the same symbol and account, where StrategiesMonitor is also running.
13. As noted above, enabling Strategy Automation can disable the Communication Owning instance of StrategiesMonitor, due to potential trade duplications and conflicts.
14. Note that TicketManager targets specific position quantities by symbol and account. No other Strategy-generated or manual orders should be placed in effected accounts, by symbol, while TicketManager is running. *(This is unless you are attempting to aid the system in reaching a target position of StrategiesMonitor.)*
15. For questions, see instructional posts here: [QCLsolutions.com](http://QCLsolutions.com) > [Support](#) > [Add-Ons](#) > [TicketManager](#). Or contact us at [Support@QCLsolutions.com](mailto:Support@QCLsolutions.com). Access our site easily with the TradeStation commentary window, for demonstration videos, prebuilt workspaces and other solutions.

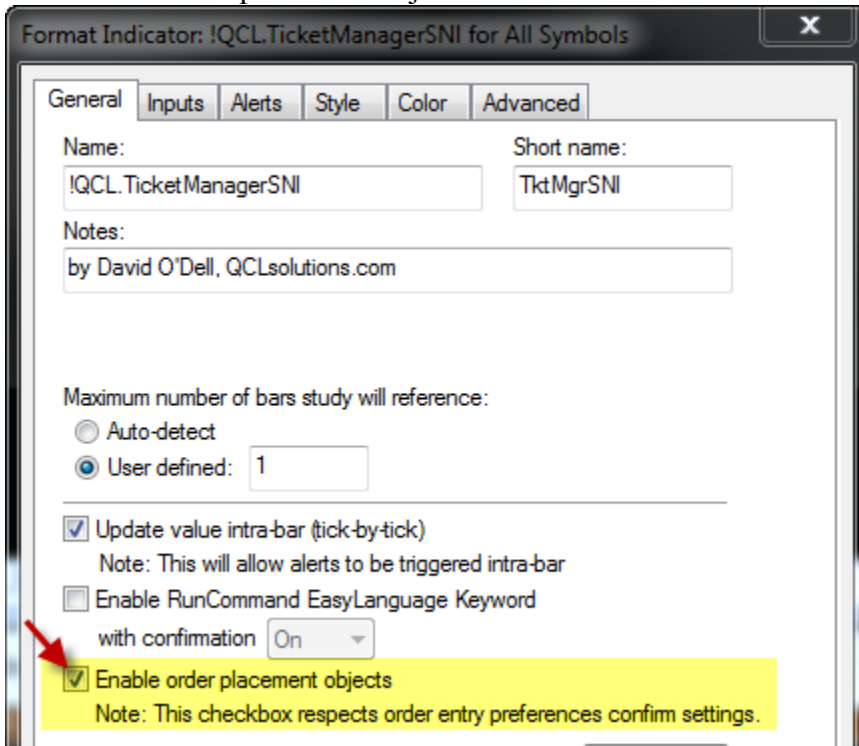
## General TradeStation Platform setup for automated Trading

1. To avoid numerous possible order confirmation and validation messages that can be burdensome and interfere with your trading, check you preferences.
  - a. File → Preferences → TradeStation Order Entry...
  - b. Under Equities, Options, Futures and Forex there are Validations & Confirms. Go to each section.

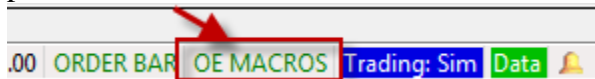


- c. At a minimum, you may want to uncheck the Confirms for “Order Bar, Macros, and Order Placement Objects”.
- d. Review Validations. Uncheck them or set them to levels that you feel will not interfere with your trading. (*Note that validation and confirmation dialogs raised prevent automated trading until each dialog is answered.*)

- To enable TicketManager trading in the window where applied, whether RadarScreen or a Chart, format the indicator and go to the General tab and check “Enable order placement objects” as follow:



- If you have set `Ok2CancelAllPriorsIfNoInfo = True`, then it is necessary to “Enable RunCommand EasyLanguage Keyword”. If this is done, you would most likely want to do so “with confirmation OFF”.
- If `Ok2CancelAllPriorsIfNoInfo = True` and RunCommand is Enable for TicketManager, then unlock “OE MACROS”. This can be done by clicking on the platform status bar OE MACROS indication so that it is green as shown.



## How to apply TicketManager yourself

1. Insert the StrategiesMonitor (!QCL.StrategiesMonitorSNI) indicator into each chart running Strategies to be auto-traded by TicketManager. *(If RunCommand is enabled note that when inserting the indicator to a chart it will attempt to add bar spaces to the right of the chart, if necessary, to make room for showing communication status text. This can raise a confirmation prompt to accept or reject the bar spacing change.)*
2. Format StrategiesMonitor in each chart where applied, and where you want it to issue orders to TicketManager for auto-trading. Go to the Inputs tab and set the following inputs:
  - a. Communicates\_ok1 = 1 (default)
  - b. IssuesOrderTickets\_ok1 = 1 (to allow issuing of tickets to TicketManager)Accept settings: OK
3. Insert the TicketManager (!QCL.TicketManagerSNI) indicator into a RadarScreen page, preferably where you will keep all symbols to be auto-traded by TicketManager.
4. Format TicketManager for all symbols in the RadarScreen page where applied. Go to the General tab and check “Enable order placement objects”.
5. You may also wish to “Enable RunCommand EasyLanguage Keyword with confirmation OFF”. This allows TicketManager to use order macros to cancel orders in cases where order objects may have failed. This is necessary when input Ok2CancelAllPriorsIfNoInfo = TRUE. Then unlock “OE MACROS” for the platform, so that order cancellation macros can work.
6. Go to the Inputs tab and set the following inputs:
  - a. TradingOn\_if\_1 = 1 (so it can place orders when tickets are received)
  - b. AccountIdEquity = “SIMyourequityacct” (try a simulation account, “” to auto-detected)
  - c. AccountIdFuture = “SIMyourfutureacct” (try a simulation account, “” to auto-detected)
  - d. AccountIdForex = “SIMyourforexacct” (try a simulation account, “” to auto-detected)
  - e. Currently the TradeStation TradingApp Store does not support customers saving default inputs. At some point TradingApp Store may. If there is a Default button on the inputs page, click the Default button to save the AccountId inputs. *(You may want to save the defaults without TradingOn\_if\_1 = 1, so set that to 0 if you prefer that for a default value.)*
  - f. **Caution: If you are logged into TradeStation for Live trading, auto-detected accounts can be real accounts. In such cases, TicketManager can send real orders.**Accept settings: OK
7. Save the workspace. *(It is recommended that you save any important workspaces created with your custom configurations. For charts setup to your preference, copy and paste them from saved workspaces. This is a great way to reduce day to day trading setup tasks. Visit us at [QCLsolutions.com](http://QCLsolutions.com) for pre-configured downloadable workspaces.)*
8. Set other inputs in TicketManager and StrategiesMonitor, to your preferences:

## TicketManager

Inputs	Demo Values	Notes
TradingOn_if_1p	0	<b>1 or greater to enable automated issuing of orders.</b> (After applying the indicator, go to the General tab and check "Enable order placement objects".) If TradingOn_ClickControl = TRUE, then -1 or less disables regardless. 0 disables unless enabled by a click event, 1 enables unless disabled by a click event, 2 or more enables regardless of click events.
TradingOn_ClickControl	FALSE	TRUE to be able to click enable/disable the effective TradingOn_if_1p setting. Note that the TradingOn_if_1p setting, if changed by clicks, is saved. If TicketManager is recalculated, historical click values persist. Set this input to FALSE to disable click functionality and clears click history.
CorePositionQty	0	A core position that is to be traded around by TicketManager. It is the neutral position quantity, for the symbol, by account. For instance if CorePositionQty is 5 and a Strategy creates a Ticket for 1 contract, TicketManager will target a total of 6 contracts (rather than 1).
MaxTargetPosQtyLong	-1	Negative values disable. 0 or positive inputs limit TicketManager trading to target Long position quantities no greater than the input.
MaxTargetPosQtyShort	-1	Negative values disable. 0 or positive inputs limit TicketManager trading to target Short position quantities no greater than the input.
AccountIdEquities	""	The account number for Equities trades. "" causes TicketManager to auto-detect a valid Equities Account.
AccountIdFutures	""	The account number for Futures trades. "" causes TicketManager to auto-detect a valid Futures Account.
AccountIdForex	""	The account number for Forex trades. "" causes TicketManager to auto-detect a valid Forex Account.
OverrideAccountID	""	"" will enable using one of the above accounts. Specifying a different account number will override anything entered above.
DayOnlyOrders	TRUE	TRUE for TicketManager to only place Day duration orders. Otherwise GTC orders can be placed and managed.



AfterHrsStocksOK	FALSE	TRUE if one wants to allow order to be placed after regular session hours.
AvoidRepeatRjtnNum	2	If an order has been rejected by TradeStation (for any number of reasons), this is the number of reattempts allowed for the exact order instructions. It is generally a good idea to allow at least one reattempt.
ShowOnlyQuantity	0	A show-only quantity to use with each order if applicable depending on symbol type, order type and order quantity. Set to 0 or less to disable.
MinIntrabrIntvlSecs	0.5	The minimum number of seconds allowed between code processing cycles, intrabar.
MinimumSpread4CalcMM	1	The minimum spread value between the inside bid and inside ask that is allowed for algorithmic limit order calculations that are relative to the spread. Specified in minimum move points.
MaxBApctFromLast2Chase	5	The maximum value allowed for tickets, for the percent change in bid or ask values. This is a percent change from the time an order is placed, that will trigger recalculation of algorithmic limit orders which are relative to market depth. (5 = 5% adverse move from last tick)
WaitBeforeFinalOrder	FALSE	TRUE to cause TicketManager to add a small waiting period before executing the final order of Ticket with multiple parts. This helps reduce the possibility of overshooting a target quantity because of latency in receiving order-filled messages.
VolPeriodSecsPreset	60	The number of seconds to use for measuring the most recent trading volume in the indicator's symbol. This input is a preset, used before it is populated with Ticket instructions as from the StrategiesMonitor.
MDVelocity_Enable	FALSE	TRUE to enable use of volume weighted rolling period averages of incoming bid and ask quotes from the MDVelocity product, through mdvPublisher.
mdVelocityMS	5000	The rolling period in milliseconds, to reference in MDVelocity published bid and ask data from mdvPublisher.

mdvMaxPrcntFromClose	5	The maximum percentage from the last close of incoming quotes, to reference in MDVelocity published bid and ask data from mdvPublisher.
mdvMaxLatentMS	10000	The maximum latency in milliseconds of incoming quotes, to reference in MDVelocity published bid and ask data from mdvPublisher.
ABrDisable_ForTickets	FALSE	TRUE enables TicketManager to disable/re-enable AutoBrackets (e.g. abfStrategyTrader) trading when managing tickets that could conflict.
qcSAS_OnStopRunCmd	FALSE	TRUE for the qcStrategyAccountStop system (included with the TicketManager product) to be able to automatically stop TicketManager trading when the qcStrategyAccountStop system cancels orders and/or closes positions according to user specified account protection limits.
HoldOrdIfPercentAway	25.0	If an order's limit price unmarketably exceeds or equals the input percent away from market price, the order will not be placed. It will then be held until market price is within the percent away.
HOILA_PrcntElseMMpts	TRUE	True for HoldOrdIfLimitPrAway as a percentage. False to specify as a number of minimum moves points away from market price.
HOILA_TrimXingLimits	TRUE	True to trim limit orders that cross market more than HoldOrdIfLimitPrAway. False to place such limit orders without trimming.
Ok2CancelAllPriorsIfNoInfo	FALSE	TRUE allows TicketManager to cancel all prior orders for the Symbol, if it cannot find TradeStation TradeManager records about earlier orders. (Typically order records are found and the correct orders are adopted by TicketManager. But, if in doubt, TRUE allows TicketManager to clear the slate.
IgnoreOrdersBeforeDate	0	In some cases TradeStation order records can be corrupt. With this input set to a valid Date greater than 0, order records after that Date, and the Time set below are ignored. This can bypass potential problems with corrupt TradeStation records. (0 or less disables the feature. Use the <b>TA_TicketManager_CrntSessStartDate</b> function to auto-set for session start.)

IgnoreOrdersBeforeTime	0	In some cases TradeStation order records can be corrupt. With this input set to a valid Time, order records after that Time, and the Date set above are ignored. This can bypass potential problems with corrupt TradeStation records. (The input is ignored if not a valid time from 0 to 2359. Use the <b>TA_TicketManager_CrntSessStartTime</b> function to auto-set for session start.)
IgnoreSessionTimes	FALSE	TicketManager is designed to prevent orders from being issued outside of session times. To bypass this protection, one may set to IgnoreSessionTimes.
PositionAcctnProbChk	FALSE	TRUE to enable position account problem checking. The system checks for position quantity changes that do not agree with order-filled events. When detected, the TicketManager instance is disabled for trading.
ShowOrderStateDetailInPlot	FALSE	TRUE to display order status details from TradeStation in plotted TktMngStatus plots.
Record_Directory	""	A file directory for TicketManager ticket and trading records to be stored in a csv file. "" to not record, create or update a file.
Record_Prefix	""	A prefix for the TicketManager records file name.
Record_Suffix	""	A suffix for the TicketManager records file name.
Record_AppendOldFile	TRUE	TRUE to append an old file and FALSE to erase existing records, starting fresh, as long as any existing file was not updated in the last 5 minutes.
Record_OrdExDetails	FALSE	TRUE to record order execution processing details along with TicketManager ticket and trading records.
DetectTimeOffset_012	2	0 or less to disable TicketManager from detecting an offset between data from TradeStation and your computer's system clock. 1 to detect a possible time zone offset, to be corrected. 2 or greater to detect any time offset, to the second. Correcting time offsets enables better management of session changes.

TimeoutMultiple	10	The multiple of a standard wait period, before a StrategiesMonitor instance is timed out, as when a chart is deleted. Tickets from timed out StrategiesMonitor instances are terminated. (A standard wait period is currently 1 second.)
ChangeToRecalculate	0	Change this input to force an indicator re-initialization without reloading data.

### StrategiesMonitor

Inputs	Demo Values	Notes
Communicates_ok1	1	1 to specify that the current instance of the indicator will communicate to aggregate positions, raise alerts, and be able to own the process of sending tickets to TicketManager. Only one instance of the indicator, by symbol and account, raises alerts and sends tickets. StrategiesMonitor automatically resolves conflicts when more than one instance is set to raise alerts and send tickets, for a symbol and account.
IssuesOrderTickets_ok1	0	If Communicates_ok1 = 1, then 1 allows sending tickets to TicketManager.
UseFormSettings_012	2	0 or less to not use the input form, using defaults below. 1 to use saved input form settings without launching the form. 2 or more to use the input form.
SymbolSettings_012	0	1 to use the indicator's "Base study on" symbol, for identifying the input form's saved settings. 2 to use the symbol root (such as CL for @CLZ18). 0 or other values for more generally saved settings.
IntervalSettings_ok1	0	1 to use the indicators "Base study on" bar interval in identifying the input form's saved settings. 0 or other values for more generally saved settings.
CustomSettingsID	""	A custom identification string to further use in identifying the input form's saved settings.
DockRight_ok1	0	The input form is docked on the left of the chart where applied, by default. Set this input to 1 to dock on the input form on the right.

TradeStartPriceOverride	0	When StrategiesMonitor begins, it may create a TicketManager ticket to set the target account to the correct number of contracts. It may use the last reported strategy price or XavgTics price to accomplish this. This input allows the user to override the other possible limit price values applicable to trade-on-indicator-start situations.
TradePriceFunctionOverride	0	For tickets to TicketManager: 0 deactivates. One can use Close, AvgPrice, or other Series functions to calculate limit price inputs to the TicketManager trading algorithms.
TkM_LmtMultiplier	1	Use to adjust strategy limit prices as when trading TkM_Symbol. Default is 1. 0 disables the feature. When trading TkM_Symbol, one may want to input TkM_Symbol as data2 and specify series calculations like: ((Close Data2)/(Close)).
TkM_QtyMultiplier	1	Use to adjust strategy quantities as when trading a proxy symbol input with TkM_Symbol. Default is 1. 0 and negative values disable the feature. When trading TkM_Symbol, one may want to input TkM_Symbol as data2 and specify series calculations like: ((Close*BigPointValue)/((Close*BigPointValue) Data2)).
CommStatusColor	Yellow	Chart highlighting color
CommStatusColorFlash	White	Chart alternating highlight color
CommStatusColorRadarRow	White	Color used for communication status plots in RadarScreen
CommStatusColorMatch	Yellow	Color used with match conditions
CommStatusColorMisMatch	Cyan	Color used with mismatch conditions
CommStatusColorBuy	Green	Color used with Buy tickets
CommStatusColorSell	Red	Color used with Sell tickets
CommStatusColorAuto	Yellow	Color used when TicketManager trading is set to Auto-Trade.
CommStatusColorManual	Magenta	Color used when TicketManager is not Auto-Trading, and some level of manual operation is necessary to trade.
ChangeToRecalculate	0	Change this input to force an indicator re-initialization without reloading data.
<b>Default values below, used when saved settings are not available, or when UseFormSettings_012 = 0</b>		

_TkM_Symbol	""	Use "" for the default tradable root of the application symbol. Otherwise, the tradable root of this symbol input is used along with the effective AccountID for sending tickets to TicketManager for execution.
_TkM_AccountID	""	Use "" for the chart's default account like the keyword <b>GetAccountID</b> which may also be used. Otherwise specify the AccountID to be used for the symbol, between quotes.
_MisMatchSecsToCheck	1	The number of seconds between indicator checks for Aggregate Strategy to TradeManager Position records to have a mismatch. Mismatches can create trade tickets for TicketManager and can create alerts.
_MisMatchSecsToTrade	3	The number of seconds between indicator checks for mismatches, to create trade tickets for TicketManager.
_MisMatchSecsToAlert	30	The number of seconds between indicator checks for mismatches, to create alerts.
_WaitForRealTimeToStart	FALSE	TRUE to force the indicator to wait for a real-time tick before it can begin detecting mismatches, creating alerts and sending tickets to TicketManager.
_EquityCrvTrade	FALSE	TRUE to enable aggregation for trading with TicketManager, based on Equity performance in the chart Strategy.
_EqHiLength	0	The look-back number of trades defining a period of highest equity performance which must be met or exceeded to begin trading with TicketManager.
_EqLoLength	0	The look-back number of trades defining a period of lowest equity performance at or below which trading would stop.
_EqHiLenMin	0	When lower than EqHiLength, EqHiLenMin is the minimum number of equity points to record before the high equity band can be used to enable equity-curve trading.
_EqLoLenMin	0	When lower than EqLoLength, EqLoLenMin is the minimum number of equity points to record before the low equity band can be used to disable equity-curve trading.

_StartEqTrading	TRUE	TRUE to initially enable aggregation for trading with TicketManager, before Equity performance rules are satisfied.
_ExitImmediate	FALSE	When equity-curve trading, and equity-curve conditions disable trading, ExitImmediate = TRUE, signals to close the simulated strategy position without waiting for a strategy exit order.
_EnterImmediate	FALSE	When equity-curve trading, and equity-curve conditions enable trading, EnterImmediate = TRUE, signals to match the simulated strategy position without waiting for a strategy entry order.
_ShowEquityCrvTotal	FALSE	TRUE to plot the simulated equity performance of the chart Strategy if Equity-Curve trading rules are followed.
_EqPointTypes_012	2	For tracking of equity-curve points used for equity-curve trade filtering and recording to file, 0 specifies all bar closes with trades or a position, 1 specifies exit trades only, and 2 or otherwise specifies entry and exit trade ticks.
_Record_Directory	"C:"	The directory of the Record file. This needs to be set for Records to happen. For example, the directory could be "c:\records".
_Record_Prefix	""	A string to be added to the front of the Record file name written in the Record_Directory.
_Record_Suffix	""	A string to be added to the end of the Record file name written in the Record_Directory.
_Record_Extension	"csv"	An extension for the Record file name written in the Record_Directory.
_ForexLotSize	1	If the symbol is a Forex instrument. This is the number of contracts per lot used with TradeStation orders. The greatest value specified in linked charts for each trading symbol, is used by TicketManager. If for every 1 quantity traded by strategies, one wants 10000 Forex units traded, then one may specify 10000 in this input.
_TradeLastStratOrXavgTics_0o1	0	For tickets to TicketManager: If 0 (default) then the limit price of tickets is based on the last strategy fill price of aggregated signals. Else if 1, then limit price inputs to the TicketManager trading algorithms are based on an exponential moving average of the last NumXavgTicks of ticks.

_NumXavgTicks	3	For tickets to TicketManager: If TradeLastStratOrXavgTics_0o1 = 1, this sets the number of ticks used for the exponential moving average of price, used as the limit price of tickets.
_TkM_LimitSlippageMM	1	For tickets to TicketManager: The slippage allowed automatically with each TicketManager algorithmic trade based on a limit price. The input expresses a multiple of minimum price moves for the symbol.
_TkM_IfTouchOffsetMM	0	For tickets to TicketManager: An If-Touched offset which the market must cross before a limit order becomes active. The input expresses a multiple of minimum price moves for the symbol. A positive If-Touched offset represents an offset away from the market. The offset is relative to an algorithmic limit price or Last market price, whichever is further from the market. This requires the market to move down to trigger a buy limit or up to trigger a sell limit order. (Note that TradeStation does not permit resulting If-Touched prices to equal or cross beyond the current market, since that would cause immediate order activation. A zero value disables If-Touched order types.)
_TkM_BAchgChkSecs	30	For tickets to TicketManager: The number of seconds between checks for changing market depth relative to last price when a ticket trade is first created. With those checks, if market depth has changed more than TkM_PrcntBAchg2Chase percent, then associated algorithmic prices are recalculated.
_TkM_PrcntBAchg2Chase	1	For tickets to TicketManager: The percent change in bid or ask values, from the last price when an order is placed. Such a percent change will trigger recalculation of algorithmic limit orders which are relative to market depth. (0.1 = 0.1% adverse move)
_TkM_RefillSecs	0	For tickets to TicketManager: The number of seconds between checks for ticket order executions, before a new ticket order will be placed. This is a minimum amount of time slicing for a ticket split into multiple orders, in order to reduce possible market impact of trying to move too much volume in a short period of time.



_TkM_RefillSecsTot	0	For tickets to TicketManager: An additional specification used with RefillSecs if 0, to set its value based on the number of expected ticket slices. If RefillSecs is greater than 0, a RefillSecsTot value greater than 0 is used to determine ticket slices, instructing TicketManager to adjust slice quantities accordingly between MaxQtyPerOrder and MinQtyPerOrder.
_RandomQtyPrcnt	50	For tickets to TicketManager: 0 for no randomization to 100 for full randomization of any calculated slice quantities, when both RefillSecs and RefillSecsTot are specified.
_RandomRefSecsPct	50	For tickets to TicketManager: 0 for no randomization to 100 for full randomization of RefillSecs between placement of slice orders of a ticket.
_TkM_MktOrderENTRY	FALSE	For tickets to TicketManager: TRUE to execute orders that create or increase the size of an asset position, with market orders. FALSE to use limit orders and more strictly control potential spread-crossing slippage.
_TkM_MktOrderEXIT	FALSE	For tickets to TicketManager: TRUE to execute orders that close or decrease the size of an asset position, with market orders. FALSE to use limit orders and more strictly control potential spread-crossing slippage.
_TkM_MktOrENTRYslpPct	2.5	For tickets to TicketManager: If TkM_MktOrderENTRY = TRUE, this is an allowable estimated slippage before a market order is placed that will create or increase the size of an asset position. (1 = 1%)
_TkM_MktOrdEXITslpPct	2.5	For tickets to TicketManager: If TkM_MktOrderEXIT = TRUE, this is an allowable estimated slippage before a market order is placed that will close or decrease the size of an asset position. (1 = 1%)

_TkM_MktOrdPlaceAsLmt	FALSE	For tickets to TicketManager: If electing to send market orders, one may want to actually send limit orders for 2 primary reasons. 1) To limit slippage based on the maximum percentages in TkM_MktOrENTRYslpPct and TkM_MktOrEXITslpPct inputs (not based on the spread). 2) To be able to trade in Equity markets after-hours when TradeStation does not accept market orders.
_TkM_SpreadCrossENTRY	0	For tickets to TicketManager: The degree of algorithmic aggressiveness in executing tickets at target limit prices. 0 specifies not crossing existing spreads (negative values can also be used, for even more patient trading). 100 enables TicketManager limit orders to cross spreads (like "Hit/Take" orders), and greater than 100 allows for crossing of spreads plus (as with "Hit/Take+" orders). This setting applies to orders that create or increase the size of an asset position.
_TkM_SprdCrsENTR_finl	100	When different from TkM_SpreadCrossENTRY, TicketManager spread-cross settings start with TkM_SpreadCrossENTRY and progress toward TkM_SprdCrsENTR_finl over TkM_SprdCrsENTR_secs seconds in TkM_SprdCrsENTR_incs increments. (When set to the same value as TkM_SpreadCrossENTRY, there is no change in TicketManager spread-cross settings.)
_TkM_SprdCrsENTR_secs	30	When sending a spread-cross settings range to TicketManager, this is the number of seconds over which the spread-cross settings are to change from TkM_SpreadCrossENTRY to TkM_SprdCrsENTR_finl.
_TkM_SprdCrsENTR_incs	3	When sending a spread-cross settings range to TicketManager, this is the number of spread-cross changes to be processed over TkM_SprdCrsENTR_secs seconds in changing from TkM_SpreadCrossENTRY to TkM_SprdCrsENTR_finl.
_RandmSCrENTReaPct	50	For tickets to TicketManager: 0 for no randomization to 100 for full randomization of time for each spread-cross value during each countdown process, when employed with entry-order slices of a ticket.

_TkM_SprdCrSINTR_2Mkt	FALSE	When sending a spread-cross settings range to TicketManager, upon reaching the final value of TkM_SprdCrSINTR_finl, if not already using market orders per the TkM_MktOrderENTRY input, TkM_SprdCrSINTR_2Mkt determines if a market order should be used.
_TkM_SprdCrSINTR_Rsts	TRUE	When sending a spread-cross settings range to TicketManager, the progression from TkM_SpreadCrossENTRY to TkM_SprdCrSINTR_finl can be for the ticket instructions, or can be reset for each filled order (i.e. ticket slices). If TkM_SprdCrSINTR_Rsts is set to TRUE, then each ticket slice starts with a spread-cross setting of TkM_SpreadCrossENTRY progressing toward TkM_SprdCrSINTR_finl.
_TkM_SpreadCrossEXIT	0	For tickets to TicketManager: The degree of algorithmic aggressiveness in executing tickets at target limit prices. 0 specifies not crossing existing spreads (negative values can also be used, for even more patient trading). 100 enables TicketManager limit orders to cross spreads (like "Hit/Take" orders), and greater than 100 allows for crossing of spreads plus (as with "Hit/Take+" orders). This setting applies to orders that close or decrease the size of an asset position.
_TkM_SprdCrSINTR_EXIT_finl	100	When different from TkM_SpreadCrossEXIT, TicketManager spread-cross settings start with TkM_SpreadCrossEXIT and progress toward TkM_SprdCrSINTR_EXIT_finl over TkM_SprdCrSINTR_EXIT_secs seconds in TkM_SprdCrSINTR_EXIT_incs increments. (When set to the same value as TkM_SpreadCrossEXIT, there is no change in TicketManager spread-cross settings.)
_TkM_SprdCrSINTR_EXIT_secs	30	When sending a spread-cross settings range to TicketManager, this is the number of seconds over which the spread-cross settings are to change from TkM_SpreadCrossEXIT to TkM_SprdCrSINTR_EXIT_finl.
_TkM_SprdCrSINTR_EXIT_incs	3	When sending a spread-cross settings range to TicketManager, this is the number of spread-cross changes to be processed over TkM_SprdCrSINTR_EXIT_secs seconds in changing from TkM_SpreadCrossEXIT to TkM_SprdCrSINTR_EXIT_finl.

_RandmSCrEXITeaPct	50	For tickets to TicketManager: 0 for no randomization to 100 for full randomization of time for each spread-cross value during each countdown process, when employed with exit-order slices of a ticket.
_TkM_SprdCrEXIT_2Mkt	TRUE	When sending a spread-cross settings range to TicketManager, upon reaching the final value of TkM_SprdCrEXIT_finl, if not already using market orders per the TkM_MktOrderEXIT input, TkM_SprdCrEXIT_2Mkt determines if a market order should be used.
_TkM_SprdCrEXIT_Rsts	TRUE	When sending a spread-cross settings range to TicketManager, the progression from TkM_SpreadCrossEXIT to TkM_SprdCrEXIT_finl can be for the ticket instructions, or can be reset for each filled order (i.e. ticket slices). If TkM_SprdCrEXIT_Rsts is set to TRUE, then each ticket slice starts with a spread-cross setting of TkM_SpreadCrossEXIT progressing toward TkM_SprdCrEXIT_finl.
_TkM_MinQtyPerOrder	1	For tickets to TicketManager: The minimum quantity allowed with a ticket order, except for the final order of a ticket which could be smaller. <i>(For Forex symbols, the value, for use in TicketManager trading, is multiplied by your specified ForexLotSize.)</i>
_TkM_MaxQtyPerOrder	10000	For tickets to TicketManager: The maximum quantity allowed with a ticket order, with the idea of avoiding excessive market impact of larger orders. <i>(For Forex symbols, the value, for use in TicketManager trading, is multiplied by your specified ForexLotSize.)</i>
_TkM_PrcntVol2Trade	0	For tickets to TicketManager: The percent of trading volume over the last TkM_VolPeriodSecs seconds, that restricts the size of individual ticket orders. 0 disables the order size restriction.
_TkM_VolPeriodSecs	60	For tickets to TicketManager: The rolling-window number of seconds over which TicketManager tracks trading volume. This volume is used as a basis of calculating the maximum size of ticket orders, relative to the TkM_PrcntVol2Trade.

_IssueHeartbeatAlerts	FALSE	If TRUE here, or in any instance of the indicator for a trading symbol, then alerts are sent every MisMatchSecsToAlert seconds. With this setting, and sending alerts to an email, a customer can be continually notified that TicketManager is running and online for the trading symbol.
_HiLitelfCommOwner_ok1	0	1 to highlight the Chart that owns communications with TicketManager. Note that the communications owning instance also controls the issuing of alerts for the symbol & account group.
_ShowCommnctnStatus_ok1	1	1 to display states of the chart strategy and tickets sent to TicketManager, on the right side.
_ChartSpacesToRight	50	Spaces added to the right of the chart, upon indicator initialization, via a macro through RunCommand functionality. Note that RunCommand must be enabled through the Format Indicator – General tab.
_DetectTimeOffst_012	0	0 or less to disable StrategiesMonitor from detecting an offset between data from TradeStation and your computer’s system clock. 1 to detect a possible time zone offset, to be corrected. 2 or greater to detect any time offset, to the second. Correcting time offsets enables better management of session changes. <i>(Disable unless _TkM_TimeTest is TRUE.)</i>
_TimeoutMultiple	10	The multiple of a standard wait period, before a StrategiesMonitor or TicketManager instance is timed out, as when a chart or symbol is deleted. A timed out StrategiesMonitor will be removed from aggregation calculations. Tickets will not be sent to a timed out TicketManager. (A standard wait period is currently 1 second.)
_TkM_TimeTest	FALSE	TRUE to test communication with TicketManager. Compares time to time requested from TicketManager .

## EquityCurveOptimizer

Inputs	Demo Values	Notes
EqHiLength	0	The look-back number of trades defining a period of highest equity performance which must be met or exceeded to begin trading. (0 or less disables the feature.)
EqLoLength	0	The look-back number of trades defining a period of lowest equity performance at or below which trading would stop. (0 or less disables the feature.)
EqHiLenMin	0	When lower than EqHiLength, EqHiLenMin is the minimum number of equity points to record before the high equity band can be used to enable equity-curve trading. (0 or less disables the feature.)
EqLoLenMin	0	When lower than EqLoLength, EqLoLenMin is the minimum number of equity points to record before the low equity band can be used to disable equity-curve trading. (0 or less disables the feature.)
StartEqTrading	TRUE	TRUE to initially allow trades before there is enough data to enable or disable trades based on equity-curve settings.
ExitImmediate	FALSE	When equity-curve trading, and equity-curve conditions disable trading, ExitImmediate = TRUE, signals to close the simulated strategy position without waiting for a strategy exit order.
EnterImmediate	FALSE	When equity-curve trading, and equity-curve conditions enable trading, EnterImmediate = TRUE, signals to match the simulated strategy position without waiting for a strategy entry order.
EqPointTypes_012	2	For tracking of equity-curve points used for equity-curve trade filtering and recording to file, 0 specifies all bar closes with trades or a position, 1 specifies exit trades only, and 2 or otherwise specifies entry and exit trade ticks.
EqCrvExtending_Enable	FALSE	TRUE to store equity-curve history and filtering status, for preloading in future runs of the strategy where one may want to load the chart with less data for speed and resources conservation.

EqCrvExtending_On1stMatch	FALSE	Effective only when EqCrvExtending_Enable is TRUE, this input determines whether stored equity curve data is merged into the first equity points of a shorter chart (i.e. True), or if merging is performed at the end of a chart when stored records are complete. This latter style is the default (i.e. False) so that the stored equity curve and signal filtering of a long chart is best reflected in shorter charts accessing the data.
OKtoRecordEqPoints_if_1	0	Set to 1 to have bar-by-bar equity-curve data written to a specified Record file.
OnEqPntTyp0_EaBar	FALSE	If TRUE then with 'EqPointTypes_012 = 0' equity curve points, per the OKtoRecordEqPoints_if_1 = 1 setting, are recorded with every bar even if there is no change in equity or trades. Otherwise only new 'EqPointTypes_012 = 0' points are recorded (i.e. points with different values from their preceding point, and/or when there is a trade).
ThirdPartyFormat	TRUE	Set to TRUE to have the above OKtoRecord inputs lead to file creation with an abbreviated name and in Open-High-Low-Close-Up-Down format, in place of the default record columns. Set to FALSE to record with an unabbreviated file name and the more extensive default data.
ThirdPartyINI_012	1	For automatic creation of a 3rd-party data attributes file. 0 to not create, 1 to create new, and 2 to create new or overwrite. Otherwise defaults to not create (i.e. 0).
Record_Directory	"C:"	The directory of the Record file. This needs to be set for Records to happen. For example, the directory could be "c:\records".
Record_Prefix	""	A string to be added to the front of the Record file name written in the Record_Directory.
IncludeSymbol_if1_root2	2	1 to have the base data symbol as part of the Record file name. 2 for the root symbol.
IncludeInterval_if_1	1	1 to have the base data stream bar interval as part of the Record file name.
IncludeName_if_1	0	1 to have the EquityCurveOptimizer name as part of the Record file name.
IncludeID_if_1	0	1 to have the application id as part of the Record file name.
Record_Suffix	""	A string to be added to the end of the Record file name written in the Record_Directory.

Record_Extension	"csv"	An extension for the Record file name written in the Record_Directory.
MaxFileNameLength	120	The maximum number of characters allowed in the created EqCrvExtending and RecordEqPoints files.
NotificationColor	Yellow	The color of a text screen notification that the Record file has been written.
OnOptimizingClosePosLastBar	FALSE	Set TRUE, if for optimizations you want the strategy to close an open position on the last bar. That way trading performance up to the last bar can be used to determine optimal inputs.
FastCalcOptimztns	TRUE	Set to TRUE for enhanced equity-curve optimization processing. With FastCalcOptimztns, optimization speeds can be improved over 20x.
MinutesBtwnRecalcs	0	To create a baseline of trade signals, EquityCurveOptimizer can trigger strategy recalculations periodically. MinutesBtwnRecalcs enforces a minimum number of minutes allowed between recalculations, to avoid unwanted recalculation loops. (0 or less disables the feature.)

### EnforceStratStartDT

Inputs	Demo Values	Notes
MaxEnforceRepeats	5	The maximum number of repeat recalculations allowed without a change in the earliest strategies start date and time calculations. (0 allows only one recalculation, and no repeats, for the same earliest strategies start date and time.)
EnforceWaitSecs	5	A number of seconds to wait before causing chart strategies to recalculate, enforcing an initially detected start date and time. (0 or less causes immediate recalculations when initially detected strategies start date and time calculations become absent.)
RC_FirstELDate	0	The First Date of the Chart's range to request with a macro command if the input EL-format Date is earlier than the first available date in the Chart's range and any prior requested First Date for the specific chart. <i>(Requires RunCommand to be enabled in Properties for All Strategies. 0 or less disables the features.)</i>



StartDates_Display	TRUE	TRUE to display the first bar date and strategy starting calculation bar date, as text in the upper-left corner of the chart.
StartCalcBar_Mark	TRUE	TRUE to mark the strategy starting calculation bar with a dotted vertical line.
StartColor	Yellow	The color of text and dotted lines used with StartDates_Display and StartCalcBar_Mark.

9. Note that TicketManager status is provided in the first plot column of RadarScreen. Extended information is in parentheses. (NRT) indicates that there is currently No Real-Time data for the particular symbol.
10. Additional resources for TicketManager:
  - a. EquityCurveOptimizer is a companion strategy intended for use with Strategy Optimizations in order to tune equity-curve trade settings. Primarily the purpose is to tune the EqHiLength and EqLoLength settings. For a video demonstration of using the optimizer visit [TicketManager at QCLsolutions.com](http://TicketManager at QCLsolutions.com).
  - b. EnforceStratStartDT is a companion strategy that enforces the start date of strategies running in the same chart. If the data of a chart is refreshed, data is received from TradeStation in packets. The EnforceStratStartDT companion strategy can cause recalculations to use new data as it arrives. Recalculations are triggered by real-time ticks, so live data is necessary. For a video demonstration, visit [TicketManager at QCLsolutions.com](http://TicketManager at QCLsolutions.com).

### Demonstration EL

Below is an EL example of using tmSignals with the TicketManager system. The example trades like the !QCL.DemoStrategy which is simply a combination of two built-in TradeStation strategies, Custom LE and Custom SE. The example is used in the bottom-right RadarScreen window of the TradingApp Store launched demonstration workspace, with multiple bar intervals of QQQ for aggregate signal trading:

---

```
{ !QCL.DemoStrategy_tmSignals indicator for instructions }
Using elsystem;
```

Inputs:

```
string TicketSymbol(""),
string TicketAccount(""),
int64 QtyForAggregation(1),
bool LECondition(C < O),
bool SECondition(C > O);
```

Variables:

```
intradaypersist string vTicketSymbol(""),
intradaypersist string vTicketAccount(""),
```

```

intrabarpersist double LastTargetPrice(0),
intrabarpersist double TargetBuyPrice(0),
intrabarpersist double TargetBuyPrDateTime(0),
intrabarpersist double TargetSellPrice(0),
intrabarpersist double TargetSellPrDateTime(0),
intrabarpersist int64 TargetBuyQty(0),
intrabarpersist int64 TargetSellQty(0),
intrabarpersist int rAggregatedApps(0),
intrabarpersist double oActiveTargetPrice(0),
intrabarpersist int64 oAggregatedQuantity(0),
intrabarpersist double oActiveTargetBuyPr(0),
intrabarpersist double oActiveTargetBuyDT(0),
intrabarpersist double oActiveTargetSellPr(0),
intrabarpersist double oActiveTargetSellDT(0),
intrabarpersist double oLastUpdatedDT(0),
intrabarpersist int oNumAggContributors(0);

```

Once Begin

```

If TicketSymbol = "" then vTicketSymbol = Symbol;
If TicketAccount = "" then vTicketAccount = tmGetAccount(Category);
Condition0 = True;

```

End;

if BarStatus(1) = 2 then begin

```

{Buy ( "CustomLE" ) next bar at market ;}
If LECondition then begin //Entry
    LastTargetPrice = AvgPrice;
    TargetBuyPrice = LastTargetPrice;
    TargetBuyPrDateTime = tmSignalStamp(false);
    TargetBuyQty = QtyForAggregation;
end else begin //Exit
    LastTargetPrice = AvgPrice;
    TargetBuyPrice = LastTargetPrice;
    TargetBuyPrDateTime = tmSignalStamp(false);
    TargetBuyQty = 0;

```

end;

```

rAggregatedApps =
    tmSignals( vTicketSymbol,
              vTicketAccount,
              TargetBuyPrice,
              TargetBuyPrDateTime,
              1, TargetBuyQty,
              oActiveTargetPrice,
              oAggregatedQuantity,

```

```

        oActiveTargetBuyPr,
        oActiveTargetBuyDT,
        oActiveTargetSellPr,
        oActiveTargetSellDT,
        oLastUpdateDT,
        oNumAggContributors);
    Once Condition0 = False;
end;

if BarStatus(1) = 2 then begin
    {Sell Short ( "CustomSE" ) next bar at market ;}
    If SECondition then begin //Entry
        LastTargetPrice = AvgPrice;
        TargetSellPrice = LastTargetPrice;
        TargetSellPrDateTime = tmSignalStamp(false);
        TargetSellQty = -QtyForAggregation;
    end else begin //Exit
        LastTargetPrice = AvgPrice;
        TargetSellPrice = LastTargetPrice;
        TargetSellPrDateTime = tmSignalStamp(false);
        TargetSellQty = 0;
    end;
    rAggregatedApps =
        tmSignals( vTicketSymbol,
            vTicketAccount,
            TargetSellPrice,
            TargetSellPrDateTime,
            -1, TargetSellQty,
            oActiveTargetPrice,
            oAggregatedQuantity,
            oActiveTargetBuyPr,
            oActiveTargetBuyDT,
            oActiveTargetSellPr,
            oActiveTargetSellDT,
            oLastUpdateDT,
            oNumAggContributors);
    Once Condition0 = False;
end;

// Optional - If tmSignal is not called above, then simply get Aggregate information
Once If Condition0 then begin
    //rAggregatedApps is a snapshot of oNumAggContributors
    rAggregatedApps =
        tmSignals( vTicketSymbol,

```

```

        vTicketAccount,
        0, 0, 0, 0,
        oActiveTargetPrice,
        oAggregatedQuantity,
        oActiveTargetBuyPr,
        oActiveTargetBuyDT,
        oActiveTargetSellPr,
        oActiveTargetSellDT,
        oLastUpdateDT,
        oNumAggContributors);
end;

//Display Targets and ID
If 0 < LastTargetPrice then Begin
    Plot1(LastTargetPrice, "TargetPrice");
    Plot2(TargetBuyQty, "TargetBUYQty");
    Plot3(TargetSellQty, "TargetSELQty");
    Plot4(TargetBuyQty+TargetSellQty, "TargetQty");
End;
Plot11(vTicketSymbol + "_" + vTicketAccount, "ID");
//Display Aggregated StrategiesMonitors and non-zero contributing tmSignals
Plot12(oNumAggContributors, "AggregApps");
If 0 < oNumAggContributors and 0 < oActiveTargetPrice then Begin
    Plot21(oActiveTargetPrice,"TargetPrice");
    Plot22(oAggregatedQuantity,"Quantity");
End;

```

---

For additional examples visit us at: [TicketManager at QCLsolutions.com](http://TicketManager at QCLsolutions.com)