Strategy Studio: Inline Pre-Trade Risk Layer

Strategy Studio servers support inline risk constraints via a simple configuration file, *risk_constraints.csv*. This file should be placed in the server's main executable directory. The server applies the constraints at the level of the strategy instance. The configuration file allows (but does not require) default values for the constraints, and these values can be overridden per strategy instance and/or per symbol (when symbol specific values are meaningful). Revisions to this file must occur prior to starting the server to take effect.

Any request to *trade_actions->SendNewOrder(...)*or *trade_actions->SendCancelReplaceOrder(...)*that violates a constraint will yield a return value of *TRADE_ACTION_RESULT_REJECT_RISK*; this happens prior to the point the order is created and passed to the execution handler. In the event a risk check fails, the server logs notice of the occurrence and additional descriptive data about the violation to the processor log file and the strategy log panel of any connected client interface, with LOGLEVEL_ERROR.

The content of the *risk_constraints.csv* file should be lines containing:

strategy_instance_name, symbol, constraint_name, limit_value

The first two fields may be left blank to provide default values: If the strategy_instance_name is blank, this will be the default limit for any strategy that is not specifically configured; similarly if symbol is blank, this will be the default limit for any instrument that is not specifically configured. Note if there are overlapping strategy/symbol defaults, such as:

strat1,,position_size,200
,SPY,position_size,800

the strategy level value (eg 200) takes precedence when strat1 trades SPY, unless there is a separate line setting:

strat1,SPY,position_size,800

Supported Risk Metrics

The following list details the behavior of the various *constraints_name*'s recognized by the server. *limit_value* for all these metrics should be non-negative numbers.

- **position_notional**
  - Limits the potential absolute notional exposure of any position, inclusive of existing working order size. Specifically, it blocks any new long(short) notional order, or CancelReplace which increases the size of an existing order, if the new notional, added to the existing long(short) working notional, added to the existing position notional, exceeds +(-)limit_value. Allows symbol specific limit_values.
- **position_size**
  - Similar to position_notional, but using size (in shares or contracts) rather than notional values. Allows symbol specific limit_values.
- **order_notional**
  - Limits the absolute maximum notional size of an individual order. Validated on new orders and CancelReplace requests which would increase the quantity of an existing order. Allows symbol specific limit_values.

- **order_quantity**
  - Sets the maximum quantity for an individual order. Allows symbol specific limit_values.
- **order_rate**
  - Sets the maximum number of outbound new orders per second.
- **order_quantity_rate**
  - Sets the maximum number of new outbound order quantity per second. Both the quantity associated with new orders and increases in quantity from CancelReplace requests decrement the allowed quota, whereas CancelReplaces which reduce quantity are always permitted.
- **message_rate**
  - Similar to order_rate, but includes CancelReplace requests.
- **portfolio_day_pnl**
  - If the portfolio's day_pnl falls below *-limit_value*, the server will block any new order, or any request to increase the quantity of an existing order, unless the new order size would work to liquidate the current position. Liquidating orders will only be allowed when the new size, combined with the existing working size in the same direction, would not risk taking the position past flat.
- **portfolio_total_pnl**
  - Similar to portfolio_day_pnl, but with reference to the portfolio's cumulative total_pnl.
- **portfolio_gross_notional**
  - The server will block any new order, or any request to increase the quantity of an existing order, if portfolio long_notional+abs(short_notional) is above *limit_value*, or if the new quantity, in conjunction with the symbol's pre-existing open order notional on the same side, risks increasing the position's absolute notional by more than the remaining space within the limit. Note this limit is liberal in the sense that it does not incorporate working notional sizes in other symbols against the limit.