Strategy Studio: TTExecutionHandler (Execution Handler)

TTExecutionHandler is Strategy Studio's FIX interface to TT's FIX Adapter order entry gateway. The adapter is currently certified for execution of futures and multi-leg instruments on the CME, CFE, Eurex and ICE exchanges.

This execution handler is enabled by adding the following setting to the server's main configuration file:

> **EXECUTION_HANDLER_MODE=TTExecutionHandler**

Session Level Settings

Session-level settings are specified in the file execution_handler-conf/tt_fix.config. Notable settings include:
> **StartDay=**_Sunday_
> **EndDay=**_Sunday_
>> • Start and End Day are required if a weekly session is configured with TT.
> **StartTime=**_19:00:00_
> **EndTime=**_17:00:00_
>> • Required. Sets the session period defining sequence number reset.
> **SocketConnectHost=**_host_
>> • Required. Target host IP.
> **SocketConnectPort=**_port_
>> • Required. Target host port.
> **ReconnectInterval=**_seconds_
>> • Reconnect wait period for a disconnected session. This should not be set lower than 60.
> **SenderCompID=**sendercompid
>> • Required; issued by TT.
> **TargetCompID=**targetcompid
>> • Required; issued by TT.
> **UseDataDictionary=Y**
> **DataDictionary=** execution_handler-conf/TT_FIX42.xml
>> • Required. May be obtained from TT's FIX Adapter support page.
> **Password=**_password_
>> • Required. Session password provided by TT.
> **StrategyStudioAccount=a**_ccountName_
>> • Required. Sets the Strategy Studio account that will be associated with this session.
> **Account=**_tag1value_
>> • Required. The value will be passed through tag 1 on each order.
> **AttributeExternalOrdersTo=**_strategyName_
>> • Optional. If a strategy named strategyName exists, external fills will be sent to this strategy. If tag 16105 ('Order Tag' field in X_TRADER) contains an instance name the fill will be routed to that instance instead.

Message validation should be disabled by setting **_ValidateUserDefinedFields=N_** and **_ValidateFieldsOutOfOrder=N_**.

Custom Order Parameters

Most order properties are set using OrderParams's fields. Order properties that are not included in OrderParams can be set via OrderParams's custom_params member, which is a std::vector<boost::any>. The values are expected to be of type std::pair<int, std::string>, with the first specifying the FIX tag number and the second specifying the string value for the FIX field. A full description of supported FIX tags is available on TT's FIX Adapter documentation site.

OrderParams's custom_params may be used on both new orders and replaces. Note that custom_params are not automatically cleared after submission of a new order; thus, care should be taken that the desired values are present in custom_params if the OrderParams is copied from an existing object.

Reference Data

Reference data should be produced on a regular basis in order to ensure correct price representation, price rounding, PnL calculation, and symbol translation.

The TTReferenceDBPopulator produces a tt_security_ids.csv that maps the LSS symbol to the FIX fields representing that symbol and the TT gateways available to trade it. The file also includes a multiplicative factor that is applied to each price as it is read from the FIX connection. This price factor defaults to 1.0 but can be set via the tt_price_factor_overrides.csv file. Note that TT may have global price adjustments configured on their side.

TTReferenceDBPopulator uses the following files:
> **tt_fix.config**
  - Required. This has the same form as the main tt_fix.config above, but should be in the same directory as TTReferenceDBPopulator.
  - **StrategyStudioAccount** should be set to "**PriceSession**".
  - **Account** and **AttributeExternalOrdersTo** are not used.
  - **ResetOnLogout** and **ResetOnDisconnect** should be set to "**Y**".
  - **SecurityDefinitionList** should be a comma separated list of <TTExchange|ProductType> pairs.
> **tt_price_factor_overrides.csv**
  - Format: Exchange,ProductCode,SymbolRegex,InboundPriceAdjustmentFactor
  - Example: "CME,RSV,RSV\u\d,.01" will set price factors for all RSV outrights to 0.01 (e.g., RSVH7).

Symbology

Instrument symbols are expected to match the first column from tt_security_ids.csv (typically this is the exchange's symbology). Note that ICE's symbols are constructed by combining the Product Code and exchange symbol; refer to tt_security_ids.csv for examples. Spryware's futures symbols are automatically converted, i.e., the forward slash is stripped and futures contracts in the form "/XX YYM" are converted to "XXMY".