Strategy Studio: TextTickReader (Backtesting Market Data)

TextTickReader is a data reader available to Strategy Studio's back testing servers. Its primary purpose is to facilitate automated strategy regression testing against pre-canned data scenarios involving common tick types. It is not intended as a high performance back testing solution.

TextTickReader may be selected as a backtesting server's data source by setting the following variable in backtester_config.txt:

- **BACKTEST_SOURCE=TextTickReader**

The following variables may be added to the server's main configuration file to customize the behavior of the adapter:

- **TEXT_TICK_DIRECTORY=*path***
  - Sets the directory to search for the text tick files. If omitted, defaults *./text_tick_data/*
- **TEXT_TICK_FIELD_SEPARATOR=*char***
  - Optional, defaults to , (comma). Specifies a character to use between fields. Should be a printable character or *\t*
- **TEXT_TICK_FEED_CONFIG_FILE**
  - Path of an optional file to configure the 'best' feed type per market center the reader should expect. Default filename is *./text_tick_feed_config.csv*. Format of the file matches the format of the server's preferred_feeds.csv config file (see Getting Started Guide). If not specified, the adapter will behave as if it expects CONSOLIDATED to be the best source of both trades and quotes.
- **TEXT_TICK_COMPRESSED=*true/false***
  - Optional, defaults to false. If set to true, the adapter will look for the tick files to be gzip compressed, with the filenames having an addition ".gz" trailer.

Text Tick files should be structured as a file per symbol per day, using the following naming convention

tick_*SYMBOL_YYYYMMDD*.txt

Ticks in these files should be ascending in time, with following event types and fields supported (note lines with any formatting errors will be skipped, with parsing errors reported to the server's *logs/BacktesterMDSource* log files):

Trades

Line Format:
COLLECTION_TIME,SOURCE_TIME,SEQ_NUM,TICK_TYPE,MARKET_CENTER,PRICE,SIZE[,FEED_TYPE,[SIDE[,TRADE_COND_TYPE,TRADE_COND]]]

- **COLLECTION_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to when data arrived
- **SOURCE_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to the timestamp sent by the data source
- **SEQ_NUM** is a uint64_t which allows deterministic ordering when merging multiple symbol files where the date and time have ties.
- **TICK_TYPE** should be set to *T* for trade ticks
- **MARKET_CENTER** identifies the market center which reported the trade. Values should be strings corresponding to Strategy Studio's MarketCenterID enum, eg "NYSE"
- **PRICE** represents trade price
- **SIZE** represents the number of shares/contracts traded

- **FEED_TYPE** (optional) integer identifying what type of feed the trade tick is from: 1 for consolidated, 2 for direct, 3 for depth (defaults to 1)
- **SIDE** (optional) indicates which side was the liquidity remover. -1 for SELL, 0 or empty string for UNKNOWN, 1 for BUY.
- **TRADE_COND_TYPE** (optional) character identifying how the trade_cond field should be interpreted. Currently available options are:
  - 'S' for SIAC CTS sale conditions; TRADE_COND should be a string of up to four CTS sale conditions.
  - 'U' for UTP UTDF sale conditions; TRADE_COND should be a string of up to four UTDF sale conditions characters.
  - 'O' for OPRA sale conditions; TRADE_COND should be an OPRA sale condition character
  - 'L' for LSE trade conditions; TRADE_COND should be a 2 character string of the LSE trade condition abbreviation (eg "AT")
- **TRADE_COND** (optional) string identifying sale conditions

Quotes (BBO)

Line Format:
COLLECTION_TIME,SOURCE_TIME,SEQ_NUM,TICK_TYPE,MARKET_CENTER,BID_PRICE,BID_SIZE,ASK_PRICE,ASK_SIZE[,FEED_TYPE, [,QUOTE_COND_TYPE,QUOTE_COND]]

- **COLLECTION_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to when data arrived
- **SOURCE_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to the timestamp sent by the data source
- **SEQ_NUM** is a uint64_t which allows deterministic ordering when merging multiple symbol files where the date and time have ties.
- **TICK_TYPE** should be set to Q for quote ticks
- **MARKET_CENTER** identifies the market center which reported the quote. Values should be strings corresponding to Strategy Studio's MarketCenterID enum, eg "NYSE"
- **BID_PRICE**
- **BID_SIZE** should be raw share/contract count, not number of lots
- **ASK_PRICE**
- **ASK_SIZE** should be raw share/contract count, not number of lots
- **FEED_TYPE** (optional) integer identifying what type of feed the trade tick is from: 1 for consolidated, 2 for direct (defaults to 1)
- **QUOTE_COND_TYPE** (optional) character identifying how the quote_cond field should be interpreted. Currently available options are:
  - 'S' for SIAC CQS quote condition character
  - 'U' for UTP UQDF quote condition character
  - 'O' for OPRA quote condition character
- **QUOTE_COND** (optional) character identifying quote condition

Depth Update By Order (OrderBook data)

Line Format:
COLLECTION_TIME,SOURCE_TIME,SEQ_NUM,TICK_TYPE,MARKET_CENTER,SIDE,ORDER_ID,PRICE,SIZE[,MMID[,REASON[,OLD_ORDER_ID[,OLD_ORDER_PRICE[,PRIORITY_INDICATOR[,IS_PARTIAL]]]]]]

- **COLLECTION_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to when data arrived
- **SOURCE_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to the timestamp sent by the data source
- **SEQ_NUM** is a uint64_t which allows deterministic ordering when merging multiple symbol files where the date and time have ties.
- **TICK_TYPE** should be set to *D* for depth by order updates

- **MARKET_CENTER** identifies the market center which reported the depth update. Values should be strings corresponding to Strategy Studio's MarketCenterID enum, eg "NYSE"
- **SIDE** identifies whether the update is for the bid or ask side of the book. 1 for buy orders, 2 for sell orders.
- **ORDER_ID** uint64_t, a unique identifier for the entry in the order book.
- **PRICE** represents the price for the entry in the order book. On a cancel-replace event this should be set to the new price
- **SIZE** represents the order size associated with the book entry. Should be zero when an order is cancelled or fully executed.
- **MMID** (optional) string identifying the market participant contributing the order book entry
- **REASON** (optional, defaults to UNATTRIBUTED) integer identifying the type of order book change propagated. If used, the value for this field should correspond to a value inferable from the exchange's raw feed messages
    - 1 (UNATTRIBUTED_CHANGE): Strategy Studio will process the depth update without assumptions on whether this is a new, deleted, or revised order.
    - 2 (ADD_ORDER): Strategy Studio will process the update under assumption it's a new entry in the book
    - 3 (PARTIAL_CANCEL): Strategy Studio will process the update under assumption it's a reduction in the size of an existing order
    - 4 (FULL_CANCEL): Strategy Studio will assume an existing order should be deleted due to a cancel
    - 5 (EXECUTED): Strategy Studio will assume an existing order should be deleted due to an execution
    - 6 (EXECUTED_AT_PRICE): Strategy Studio will assume an existing order should be deleted due to an execution at a price other than the resting order's previously advertised price
    - 7 (MODIFY): Strategy Studio will update the entry for an existing order id
    - 8 (CANCEL_REPLACE): Strategy Studio will process the depth change as an atomic removal of one book order, and addition of another. Requires correct population of OLD_ORDER_ID.
- **OLD_ORDER_ID** (conditionally required when REASON = CANCEL_REPLACE) used to identify the previous order id during a CANCEL_REPLACE event
- **OLD_ORDER_PRICE** (optional) used to identify the previous order price during a CANCEL_REPLACE event
- **PRIORITY_INDICATOR** (optional, default is 0) uint64_t queue placement priority indicator if supplied by the data source. Within price level, lower values indicate higher priority in the book.
- **IS_PARTIAL** (optional, default is 0) Boolean as 0 or 1 indicating whether this is part of a batch of updates and not the last update of the batch. If so, triggering of OnQuote/OnTopQuote will be skipped.

Depth Update By Price (OrderBook data)

Line Format:
COLLECTION_TIME,SOURCE_TIME,SEQ_NUM,TICK_TYPE,MARKET_CENTER,SIDE,PRICE,SIZE[,NUM_ORDERS[,IS_IMPLIED[,REASON[,IS_PARTIAL]]]]

- **COLLECTION_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to when data arrived
- **SOURCE_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to the timestamp sent by the data source
- **SEQ_NUM** is a uint64_t which allows deterministic ordering when merging multiple symbol files where the date and time have ties.
- **TICK_TYPE** should be set to *P* for depth by order updates
- **MARKET_CENTER** identifies the market center which reported the depth update. Values should be strings corresponding to Strategy Studio's MarketCenterID enum, eg "NYSE"
- **SIDE** identifies whether the update is for the bid or ask side of the book. 1 for buy prices, 2 for sell prices.
- **PRICE** identifies the price for the level in the order book.
- **SIZE** represents the aggregate size present at the price level. Should be zero when the price level is fully depleted or falls outside of the exchange's update range.
- **NUM_ORDERS** (optional) The number of orders present at the price level, should be 0 for implied updates.
- **IS_IMPLIED** (optional default is 0), Boolean as 0 or 1 indicating whether this is an implied price level.

- **REASON** (optional, defaults to UNATTRIBUTED) identifies the reason the order book changed.
- **IS_PARTIAL** (optional, default is 0) Boolean as 0 or 1 indicating whether this is part of a batch of updates and not the last update of the batch. If so triggering of OnQuote/OnTopQuote will be skipped.

Order Book Reset

Line Format:
COLLECTION_TIME,SOURCE_TIME,SEQ_NUM,TICK_TYPE,MARKET_CENTER

- **COLLECTION_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to when data arrived
- **SOURCE_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to the timestamp sent by the data source
- **SEQ_NUM** is a uint64_t which allows deterministic ordering when merging multiple symbol files where the date and time have ties.
- **TICK_TYPE** should be set to *R* for book resets
- **MARKET_CENTER** identifies the market center which reported the book reset. Values should be strings corresponding to Strategy Studio's MarketCenterID enum, eg "NYSE"

Imbalance

Line Format:
COLLECTION_TIME,SOURCE_TIME,SEQ_NUM,TICK_TYPE,MARKET_CENTER,AUCTION_TYPE,MATCHED_QUANTITY,IMBALANCE_SIZE,IMBALANCE_SIZE2,CLEARING_PRICE,CLEARING_PRICE2,REFERENCE_PRICE

- **COLLECTION_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to when data arrived
- **SOURCE_TIME** should be UTC date time, specified in format '*yyyy-MM-dd HH:mm:ss.ffffff*' corresponding to the timestamp sent by the data source
- **SEQ_NUM** is a uint64_t which allows deterministic ordering when merging multiple symbol files where the date and time have ties.
- **TICK_TYPE** should be set to *I* for imbalance messages
- **MARKET_CENTER** identifies the market center which reported the auction update. Values should be strings corresponding to Strategy Studio's MarketCenterID enum, eg "NYSE"
- **AUCTION_TYPE** integer identifying what type of auction the imbalance update pertains to. Possible values are:
  - 1 (OPEN)
  - 2 (MARKET)
  - 3 (HALT)
  - 4 (CLOSE)
  - 5 (NONE)
  - 6 (REGULATORY)
  - 7 (IPO)
- **MATCHED_QUANTITY** Quantity of shares that are matched/paired off at the reference price
- **IMBALANCE_SIZE** Number of shares that are out of balance, normalized to a signed integer (positive for more buy orders than sell orders)
- **IMBALANCE_SIZE2** Number of shares that are out of balance based off of just market orders. Use zero if not applicable.
- **CLEARING_PRICE** Closest indicative price where imbalance would be zero
  - 'near price' for NASDAQ
  - 'clearing price' for NYSE Open imbalance and 'continuous book clearing price' for closing imbalance
  - 'indicative match price' for ARCA Book Multicast, now called 'reference price' in ARCA Book XDP
- **CLEARING_PRICE2** Provides additional indicative measure of clearing price where applicable. Use zero if not applicable
  - 'far price' for NASDAQ
  - 'closing only clearing price' for NYSE
- **REFERENCE_PRICE** Reference price corresponding to the imbalance quantities. Use zero if not applicable