MultiTradex/TMS Client API

Terminology

- **MultiTradex instance**: a running instance of MultiTradex client that has already logged in; note that TMS client API is fully compatible with that of MultiTradex.
- **User program**: third-party user program that communicates with MultiTradex/TMS client through this API.

Summary

MultiTradex API allows a third-party user program to get connected with a running instance of MultiTradex through TCP/IP sockets. At the moment, through the API the user can,

- **Submit a new regular/strategy order**
  - A regular order refers to a normal order for any of the supported exchanges.
  - A strategy order refers to an order for a strategy contract defined in MultiTradex built-in spreader engine.
- **Amend/Cancel orders**
- **Receive order/fill status updates (passive)**
- **Receive real-time price updates with full available market depth (passive)**
  - Updates will be received on contracts subscribed in MultiTradex price tabs, in live positions, or those subscribed by user program itself.
- **Create and edit strategies for build-in spreader engine**
- **Receive real-time account information updates (passive)**
- **Receive all the entire Book & Strategy collection that is configured for the TMClient user.**

Once connected to the specified TCP/IP sockets on the MultiTradex instance (please refer to section: **How to Activate the API** for more details), the user program will send commands and receive information via **text messages**. **All text messages must be encoded in UTF-8**.

How to Activate the API

The API interface is disabled by default. To activate the API listener on MultiTradex/TMS client, please create a file named "*remote.txt*" in the folder:

MultiTradex: **%appdata%\multitradex**

TMS: **%appdata%\TMClient**

(**Note that this is different from the installation folder of MultiTradex/TMS!!!**)

Open the txt file and put on the line below:

enabled=true

and then save the changes to the file.

The next time you log into MultiTradex/TMS client, in either Demo or Live mode, the API listener will fire up (to accept text commands). Make sure when the firewall asks for your permission to open up TCP ports, approve it.

The trade interface listens on TCP Port 55241

The market data interface listens on TCP port 55242

Additional Configuration Options

The following <Key>=<Value> pairs are accepted in *remote.txt*. Each of such pair must be placed on a separate line.

Key	Value
`enabled`	true or false (default to false if key not specified)
`tradeBindIP`	Bind IP for trade listener, default to 0.0.0.0 (any IP)
`tradeBindPort`	Default to 55241
`marketBindIP`	Bind IP for market data listener, default to 0.0.0.0 (any IP)
`marketBindPort`	Default to 55242
`acceptIPs`	You can specify a list of IP or IP ranges that are allowed to connect in, separated by semicolon characters. You can use wildcard characters, e.g. 192.168.0.1; 192.168.2.* Default to *.*.*.*

General Message Format

(**Important Note: all angular brackets "<>" used subsequently are not meant to be included in the actual message. They are placeholders to indicate a message component**.)

All messages, whether initiated by MultiTradex or the user program, have the following format:

<message header> <message body> <tail>

- *<message header>* is of the form

  #ddddddddd#

Where **#** is just the regular hash character, and the **ddddddddd** is a decimal representation of a 10-digit number for the length of the *<message body>* plus *<tail>* (**in bytes with UTF-8 encoding**). The number is possibly padded with leading zeroes, e.g. #0000000065#.

- *<message body>* is the actual command of the form

  <type><SOH><field_name1=value1><SOH><field_name2=value2>…

  Where the *<type>* indicates the class of the message, possibly followed by a list of fields delimited by *<SOH>* (the Unicode character corresponding to Hex value 0x01).

- *<tail>* is '\n', i.e. the line feed character.

Message Body

After the *<type>* specifier, each field is given in the form of *field_name=value*.

The following types of messages are supported by trade interface,

- ***SendOrder****<SOH>ClientOrderID=123<SOH>Contract=STRATEGY\GoldS\JUN12-JUL12<SOH>BuySell=Buy<SOH>Quantity=2<SOH>Price1=0.12<SOH>Options=C-Strategy<STX>RA001*

  SendOrder instructs the system to submit a new order. The fields are,

Field Name	Optional	Remark
ClientOrderID	N	A unique (until the user restarts MultiTradex) ID generated by the user program to identify this order in future operations, e.g. amend or cancel. User-supplied *ClientOrderID* may not start with "SYS" as it is reserved for use by the system itself.
Contract	N	The contract for which order is for, e.g. *CME\Gold\JUN12*, or *STRATEGY\GoldS\JUN12-JUL12*.
BuySell	N	Either *Buy* or *Sell.*
Quantity	N	Total lots.
OrderType	Y	One of *Market*, *Limit*, *Stop*, *StopLimit*; **Default to *Limit*.** **Note: Market orders for all Chinese exchanges together with LME will be changed to Limit orders due to a lack of support for "Market" OrderType in these exchanges. Have no worry about your market orders --- they will still work as market orders.**
Price1	Y	Limit price of a *Limit* order, or trigger price of a *Stop*/*StopLimit* order.
Price2	Y	Limit price of a *StopLimit* order.
Account	Y	Trader account(s) to be used for the order; if not specified the system will use the first account(s) in the relevant dropdown list in MultiTradex instance; for a STRATEGY contract with multiple legs, the accounts must be given in the form of *<account for leg1>|<account for leg2>…* ('|' is the vertical bar character) **Note: you could also use the TAB character to separate accounts for multiple legs as in the older version, but it is deprecated; similarly for *OpenClose* and *HedgeType* below.**
OpenClose	Y	Only needed for China contracts; can be one of *O* (for open), *C* (for close), *T* (for close today), *A* (for auto); **If not specified will Default to *A***; For a contract of DCE, can also be one of OC (for OpenClose), CO (for CloseOpen). For a contract of SHFE, can also be T (For CloseToday).

		For a STRATEGY contract with multiple legs, it needs to be specified as *<OC for leg1>|<OC for leg2>…*
HedgeType	Y	Only needed for China contracts; can be one of *S* (for speculation), *H* (for hedge), *A* (for arbitrage); **If not specified will default to *S***; For a STRATEGY contract with multiple legs, it needs to be specified as *<HedgeType for leg1>|<HedgeType for leg2>…*
AutoClose	Y	This corresponds to the "Auto Close" VIP function. The value represents the desired auto-close price when the primary order is (partially) filled.
AutoCancel	Y	This corresponds to the "Auto Cancel" function. You can specify a time in the format of **HH:mm:ss** (e.g. 23:01:00). The order will be automatically canceled at that time if not already completed.
Condition	Y	This corresponds to the conditional order function. The value needs to be of the format <Condition>|<Operator>|<Operand>, where <Condition> can be, **Bid** **Offer** **Last** **Loop** And <Operator> can be, **>=** **<=** **>** **<** Note that "Loop" doesn't require <Operator>. See two examples below, **Bid|>=|100.57** **Loop||10**
Options	Y	The optional fields/tags that you might want to attach when sending an order. Currently supported tag(s); 1. C-Strategy Example: Options=C-Strategy<STX>RA001 Note that inside Options, we separate all keys and values using <STX> which is 0x02 in ASCII.
TWAPParams	Y	Attatch this field when you want to send a TWAP or FAKAlgo order.

		Tags in TWAPParams (all tags should have numerical values unless explicitly specifiied below): 1. TotalTimeSeconds 2. SliceTimeSeconds 3. TotalLots 4. OrderSizeVariation 5. TimeSliceVariation 6. EnableFAK Only use “True” or “False” 7. InitialOffset 8. TWAPIsFAKAlgo Only use “True” or “False” 9. TWAPFAKRefillTimes 10. TWAPLastSliceRefillTimes 11. TWAPFAKOffset 12. TWAPRetryLimit 13. TWAPRetryWaitTimeSeconds 14. TWAPTimeExtension Example: TWAPParams=TotalTimeSeconds<STX>120<STX>SliceTimeSeconds<STX>20<STX>TotalLots<STX>40<STX>OrderSizeVariation<STX>0.3<STX>TimeSliceVariation<STX>0.1<STX>EnableFAK<STX>True<STX>InitialOffset<STX>1<STX>TWAPIsFAKAlgo<STX>True<STX>TWAPFAKRefillTimes<STX>1<STX>TWAPLastSliceRefillTimes<STX>0<STX>TWAPFAKOffset<STX>1<STX>TWAPRetryLimit<STX>1<STX>TWAPRetryWaitTimeSeconds<STX>5<STX>TWAPTimeExtension<STX>0.3 Note that the flags keys and values are separated by <STX> which is 0x02 in ascci. You can refer to the python demo script included in the package.

- ***CancelOrder****<SOH>ClientOrderID=123*

  CancelOrder instructs the system to cancel the order identified in ClientOrderID.

- ***AmendOrder****<SOH>ClientOrderID=123<SOH>Quantity=10<SOH>Price1=0.15*

  AmendOrder instructs the system to modify the Quantity, Price1 or Price2 of an order, identified in ClientOrderID.

  Things to notice:

  1. AmendOrder action for Chinese contracts is not supported by Chinese exchanges: DCE, SHFE, CZCE, and CFFEX. Such condition leads to a special way of amending Chinese

contracts' orders: Cancel the previous order and send a new order with amended parameters.

- ***OrderUpdate****<SOH>ClientOrderID=123<SOH>ServerOrderID=STRATEGY\ST0<SOH>Contract=STRATEGY\GoldS\JUN12-JUL12<SOH>Account=DEMO1|DEMO2<SOH>BuySell=Sell<SOH>Status=Working<SOH>Quantity=2<SOH>Price1=0.12<SOH>AmountFilled=0<SOH>AveragePrice=0<SOH>C-Strategy=RA001*

  OrderUpdate is a type of message initiated by the MultiTradex instance to notify the user program of the latest status of orders. The fields are,

Field Name	Optional	Remark
ClientOrderID	N	A unique (until the user restarts MultiTradex) ID that may be used by the user program to refer to this order, e.g. for cancel or amend. IDs starting with "SYS" are for orders initiated in MultiTradex itself rather than by the user program.
ServerOrderID	N	API ID + The order ID as shown in MultiTradex UI. (Example: CTP\1-1368046-7)
Contract	N	The contract for which order is for, e.g. *CME\Gold\JUN12*, or *STRATEGY\GoldS\JUN12-JUL12*.
BuySell	N	Either *Buy* or *Sell.*
Status	N	Could be one of *Filled*, *Cancelled*, *Rejected, Working or Queued*. **Filled** stands for this order has been filled. **Cancelled** stands for this order it canceled and hence will not take effect anymore. **Rejected** stands for this order cannot be submitted due to some reason. **Working** stands for this order is in the market and waiting to be filled by other orders. **Queued** stands for this order is now processing by Tradex Server and will be submitted to exchanges right away.
Quantity	N	Total lots.
Price1	Y	Limit price of a *Limit* order, or trigger price of a *Stop/StopLimit* order.
Price2	Y	Limit price of a *StopLimit* order.
Account	N	See relevant entry of *SendOrder*
AmountFilled	N	Lots filled so far with this order.
AveragePrice	N	The average price of the filled lots.
RelatedOrder	Y	The *ServerOrderID* of the related STRATEGY order, if any.
Error	Y	Error message, if any
OpenClose	Y	Open/Close flag, see relevant entry of *SendOrder*
HedgeType	Y	Hedge flag, see relevant entry of *SendOrder*
OrderType	N	Type of the order. Could be one of the following: Market|Stop|StopLimit|Limit
C-Strategy	Y	Strategy tag you sent together with your new order.

Things to notice:

1. When accessing the API through the TMS Client software, ***OrderType*** field will be changed to "**Limit**" when you are submitting "**Market"** orders whose contracts are from DCE, SHFE, LME or CFFEX exchanges.
   In the situation above, a submitted "**Market"** order will act as a "**Market"** order despite its ***OrderType*** is changed to "**Limit**" by API.
2. Take note that an OrderUpdate message includes two orderID fields: **ClientOrderID** and **ServerOrderID**.
   ClientOrderID field stands for the OrderID you submitted your order through TradexAPI while **ServerOrderID** stands for the OrderID used in our MultiTradex/TMS software after our software receive the new order from TradexAPI. To keep track of an order safely, it is always a good practice to take note both OrderID.

- ***FillReceived****<SOH>FillID=67890<SOH>ClientOrderID=123<SOH>ServerOrderID=STRATEGY\ST 0<SOH> Contract=STRATEGY\GoldS\JUN12-JUL12<SOH>Account= DEMO1|DEMO2<SOH>BuySell=Sell<SOH>Quantity=2<SOH>Price=0.12<SOH>FilledTime=17:0 5:38<SOH>C-Strategy=RA001*

  FillReceived is a type of message initiated by the MultiTradex instance to notify the user program of any fills (An order could have one or more fills). The fields are,

Field Name	Optional	Remark
FillID	N	The fill ID as shown in MultiTradex UI.
ClientOrderID	N	A unique (until the user restarts MultiTradex) ID that may be used by the user program to refer to this order, e.g. for cancel or amend. IDs starting with "SYS" are for orders initiated in MultiTradex itself rather than by the user program.
ServerOrderID	N	The order ID as shown in MultiTradex UI.
Contract	N	The contract for which order is for, e.g. *CME\Gold\JUN12*, or *STRATEGY\GoldS\JUN12-JUL12*.
BuySell	N	Either *Buy* or *Sell.*
Quantity	N	Filled lots.
Price	N	Filled Price.
Account	N	Trader account under which the fill is done.
FilledTime	N	Format is HH:mm:ss
OpenClose	Y	Open/Close flag, see relevant entry of *SendOrder*
HedgeType	Y	Hedge flag, see relevant entry of *SendOrder*
C-Strategy	Y	Strategy tag you sent together with your new order.

- ***PositionUpdate****<SOH>Account=PATS\DEMO1<SOH>Contract=CME\CPO\OCT13<SOH>Hedge Type=<SOH>Buys=1<SOH>Sells=3<SOH>PrevBuys=<SOH>PrevSells=<SOH>BuyAveragePrice= 0<SOH>SellAveragePrice=699*

PositionUpdate is a type of message initiated by the MultiTradex instance to notify the user program of the latest status of positions. The fields are,

Field Name	Optional	Remark
Account	N	Trader account under which the position belongs to.
Contract	N	The contract for which Trader account is for.
HedgeType	N	Currently, its default value is empty.
Buys	N	Buying lots of today.
Sells	N	Selling lots of today.
PrevBuys	N	Buying lots of yesterdays.
PrevSells	N	Selling lots of yesterdays.
BuyAveragePrice	N	If net position <= 0, the value will be 0. Otherwise, its value is Avg price in Position Tab of UI.
SellAveragePrice	N	If net position >= 0, the value will be 0. Otherwise, its value is Avg price in Position Tab of UI.
ClosedPL	N	Accumulate Profit/Loss

- ***TradeLinkStatus*** *<SOH>UpstreamID=CTP<SOH>Status=Up*

  This message is received whenever the trade connection status between the MultiTradex instance and any of the upstream providers has changed.

Field Name	Optional	Remark
UpstreamID	N	Identifier for the upstream (can be checked in trader account dropdown list for the respective upstream provider)
Status	N	Link status change: *Up* or *Down*

- ***SnapshotDone***

  This message has no fields.

  - The instant the user program connects to the trade interface of a MultiTradex instance, MultiTradex will send a complete snapshot of the current system status to the user program.
  - The snapshot consists of the latest status of all orders and all received fills, as well as link status for all upstream providers.
  - Therefore immediately in connection, the user program should expect to receive *OrderUpdate* and *FillReceived* messages for all the orders and fills, and *LinkStatus* messages (in the state they are at the time of connection).
  - When all the orders and fills have been transmitted, MultiTradex will send this special *SnapshotDone* message to the user program. Thus, any further messages the user program receives after this point are for fresh updates after the time point the snapshot was taken.

- ***CreateStrategy****<SOH>Contract=STRATEGY\ABC\EFG<SOH>PriceFormula=p1-p2<SOH>TickSize=0.01<SOH>Legs=CME\Crude Oil\JUN07|CME\Crude Oil\SEP07<SOH>Side=+|-<SOH>LotsMul=1|1<SOH>SupportMul=2|2<SOH>Tolerance=1|1<SOH>Mode=A|P*

  Create a new strategy or edit an existing one, depending on whether field *OldContract* is present. The valid fields are:

Field Name	Optional	Remark
Contract	N	A name for the desired strategy contract, e.g. *STRATEGY\ABC\EFG*. Must start with *STRATEGY* and needs to be in the format of *STRATEGY\<contract>\<date>*
OldContract	Y	Set this field to the name of the strategy you want to modify; this field must be absent if you are creating a new strategy.
PriceFormula	N	Strategy price formula, e.g. *p1-p2*
TickSize	N	Strategy tick size, e.g. *0.01*
MinActiveVol	Y	Minimum active volume, default to 1
MaxVol	Y	Max volume, default to 0 (unlimited)
Legs	N	Leg contracts separated by | (same for below), e.g. *CME\Crude Oil\JUN07|CME\Crude Oil\SEP07*
Side	N	Leg sides, e.g. *+|-*
LotsMul	N	Leg lots multipliers, e.g. *1|1*
SupportMul	N	Leg support multipliers, e.g. *2|2*
Tolerance	N	Leg tolerance, e.g. *1|1*
Mode	N	Modes, A for active and P for passive, e.g. *A|P*
SafeTicks	Y	Safe depth in ticks, e.g. *5|5*; you can also use the keyword *Disabled* (**similarly for all applicable parameters below**), e.g. *5|Disabled*; default to *Disabled* for all legs
ActiveRegion	Y	Active regions in ticks, e.g. *1|1*, default to 1 for all legs
MinMove	Y	Minimum movement in ticks, e.g. *1|1*, default to 1 for all legs
AlTicks	Y	Auto-liquidate on stop loss in ticks, e.g. *6|6*, default to *Disabled* for all legs
AlTime	Y	Auto-liquidate on time in seconds, e.g. *30|30*, default to *Disabled* for all legs
HedgeFlag	Y	Default hedge flags, e.g. *S|H*, default to S for all legs

  You will receive an *ErrorMessage* indicating the reason if your creation/modification is not accepted. Otherwise if successful, you will receive a *ContractUpdated* message as explained below.

- ***ContractUpdated****<SOH>Contract=STRATEGY\ABC\EFG*

  This message is received whenever a contract is either added or updated. Currently, you only receive this triggered by successful Strategy contract addition/update. It will have more useful fields in the later versions of API.

Field Name	Optional	Remark
Contract	N	New contract name for the strategy

OldContract	Y	The name of the old contract if you are modifying one.

- ***KeepAlive***

  This message has no fields, and API sends a keep-alive message "KeepAlive". If MT receives the message from API, it will send message "**ACKKeepAlive**" to notify Client that the Tradeconnection still works.

- ***AccountUpdate****<SOH>Account=OTC\ezio<SOH>Balance=123324<SOH>Margin=1920<SOH>Cash=40000<SOH>UnPL=21022<SOH>ClosePL=23133*

  AccountUpdate is a type of message initiated by the MultiTradex instance to notify the user program of the latest status of users' accounts. Note that AccountUpdate will be pushed to user program 1 time per second. Fields included are:

Field Name	Optional	Remark
Account	N	API ID + Trader account.
Balance	N	The balance of the above account.
Margin	N	The margin for the above account.
Cash	N	The cash for the above account.
UnPL	N	The unrealized (float) profit/loss for the account.
ClosePL	N	The close profit/loss for the account.

- ***CargillBookStrategyCollection****<SOH> C-Book=BM<SOH>C-Strategy=RA001<SOH>NextItem=true<SOH>C-Book=Aster<SOH>C-Strategy=EF001<SOH>NextItem=true<SOH>C-Book=Aster<SOH>C-Strategy=EF002<SOH>NextItem=true*

  CargillBookStrategyCollection is a type of message initiated by the MultiTradex instance to notify the user program of the latest collection of **all** Book & Strategy items.

  ①This message will be sent/pushed to the user program when a new API connection is established, before the ***SnapshotDone*** message is sent to the user program.

  ②This message will also be sent/pushed to the user program during a connection session, only when the MultiTradex instance's Book & Strategy items are modified by the server/backoffice side.

  Fields included are:

Field Name	Optional	Remark
C-Book	N	The book that a strategy tag belongs to.
C-Strategy	N	The strategy tag that a user might want to attach to his/her orders.
NextItem	N	Reserved for future usage. Currently, the value will always be true.

The following types of messages are supported by market data interface,

- ***PriceLinkStatus*** *<SOH>UpstreamID=CTP<SOH>Status=Up*
  This message is received whenever the price connection status between the MultiTradex instance and any of the upstream providers has changed. **You will also always receive these messages immediately on connecting to the market data interface indicating the present status.**

Field Name	Optional	Remark
UpstreamID	N	Identifier for the upstream (can be checked in trader account dropdown list for the respective upstream provider)
Status	N	Link status change: *Up* or *Down*

- ***Subscribe****<SOH>Contract=X\Y\Z*

  Subscribe to the specified contract and passively receive all real-time price updates on the contract. **Note that contracts subscribed with this message will not appear in UI (price tabs)**.

  You can subscribe to the same contract more than once, in which case you will only end your subscription by *Unsubscribing* it the same number of times.

- ***Unsubscribe****<SOH>Contract=X\Y\Z*
  Unsubscribe from the specified contract.

- ***PriceUpdate****<SOH>Contract=X\Y\Z<SOH>Bid=<price DOM><SOH>Offer=<price DOM><SOH>Last=<last volume>,<last price><SOH>Low=<low price><SOH>High=<high price><SOH>Open=<open price><SOH>Close=<close price><SOH>Settlement=<stl price><SOH>Total=<total volume><SOH>Timestamp=<server time stamp, HH:mm:ss>*

  Where <price DOM> is of the format:

  <volume1>,<price1>,<volume2>,<price2>…

  The prices in the DOM list is ordered in ascending order for offer, and descending order for bid.

  PriceUpdate message is a type of message initiated by the MultiTradex instance to notify the user program about the price update. Price update info will be pushed to you when you have subscribed the price of wanted contracts or you have done this inside your tradex client application. You can also subscribe contracts' price info using this API's Subscribe Functionality.

  Fields are:

Field Name	Optional	Remark
Contract	N	The contract name of this price update message.

Bid	N	Market bid volumes and corresponding prices. Highest bid price and its volume first.
Offer	N	Market offer volumes and corresponding prices. Lowest offer price and its volume first.
Last	N	Volume and price of the last transaction of this contract.
Low	N	Lowest transaction price of today.
High	N	Highest transaction price of today.
Open	N	The price of the first transaction since market open.
Close	N	The price of the last transaction before the market is closed.
Settlement	N	The weighted average price during a market's open and close.
Total	N	The total volume of all transactions during a market's open and close.
Timestamp	N	The time of this PriceUpdate message.

Things to notice:

1. Timestamp Field:
   When using **MultiTradex software**, then you are supposed to receive timestamps for all PriceUpdate message whose values are all 00:00:00.
   When using **TMS Client software**, you are supposed to receive timestamps whose values are configured in our TMS Server as a 24-hour format.
   In both cases, timestamps received are not from the exchanges.

2. Cached Price Data: This API pushes all cached price data to you once a new API connection is established. This feature, however, may cause confusion when the cached price data were the ones just before a market is closed. That is to say: if you connect to this API just after a market is re-opened, then you may receive cached PriceUpdate messages whose timestamps are near the market's last closing time rather than the market's latest open time.

- ***KeepAlive***
  This message has no fields, and API sends a keep-alive message "KeepAlive". If MT receives the message from API, it will send message "**ACKKeepAlive**" to notify Client that the Market connection still works.

Important Notes

- The communication is fully duplex. MultiTradex will in real time update status of all orders submitted through the API, from the moment the user connects to the listener port.
- All orders, whether submitted by the user program (yours or of another connected user), or from within MultiTradex UI (i.e. manual click), will be visible to the user program.
- *ClientOrderID* must be unique for each new order sent with *SendOrder* until MultiTradex is shut down and restarted; the *ClientOrderID* used by *SendOrder* cannot start with "SYS".

- By default you will receive all price updates for contracts set in UI (price tabs); you can subscribe to additional contracts with *Subscribe* message.