CbJOtsET

Cboe Japan OTS Feed

This Document

CbJOtsET.pdf Rev 1.1 – Details how to install, configure and run CbJOtsET.

Revision

18/10/2023 – 1.1 - C.C. – Final Revision
26/09/2023 – 1.0 - C.C. – Initial draft

1. Overview

CbJOtsET is an application developed by RJE Systems that provides users with a connection to Cboe Japan's (Cboe) order information, captured in real time. To run the application, the user will need a FIXDROP (TCP/IP) connection with the Cboe exchange to extract execution reports from the Cboe production / test systems.

CbJOtsET will work with the following Cboe Feeds: -

- Cboe Trade Feed
- Cboe 3rd Party Consolidated Feed

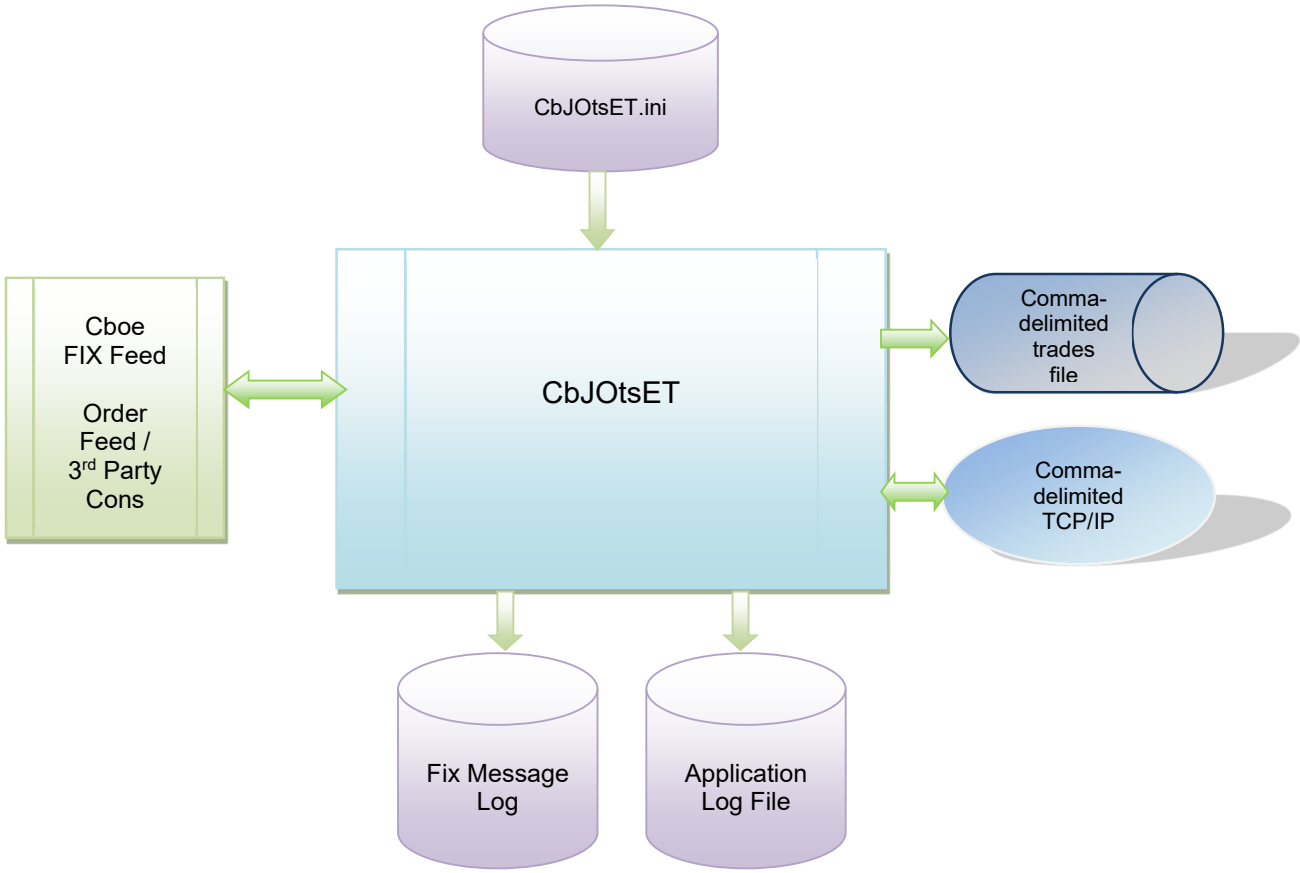


Figure 1. The CbJOtsET Production System

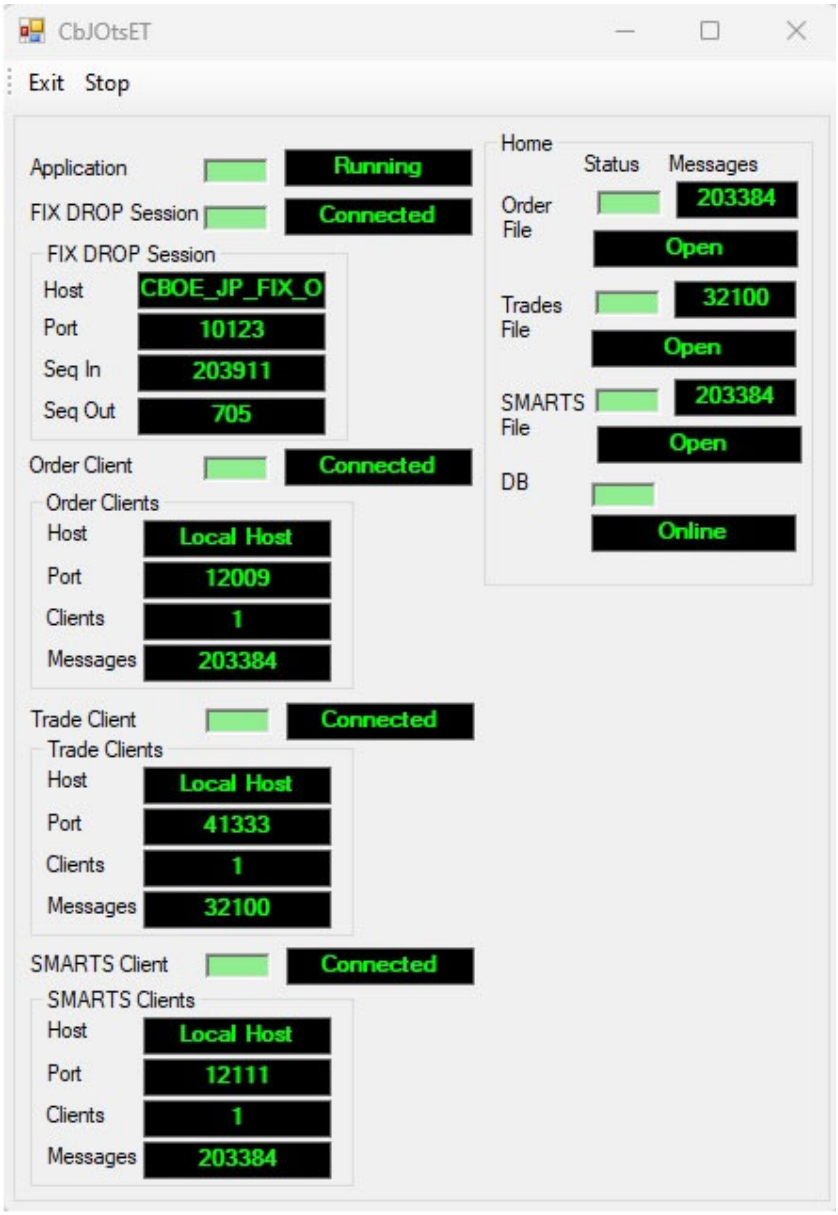
1.1 Outputs

Output Data is available as: -

- Comma-delimited Trade File (single trade side)
- Comma-delimited TCP/IP Trade Feed (single trade side)

The Comma-Delimited TCP/IP Trade Feed is similar to all other RJE's trades' products.

1.2 GUI Screen:



The application provides a GUI screen that gives a quick visual indication that everything is operating.

Typically, good status values are Green but may transit to other states during stopping and starting operations. CbJOtsET status values are indicated as follows:

Application Status: -
- Starting (Orange)
- Running (Green) – normal
- Stopping (Red)
- Hibernating (Grey) – normal overnight.

FIX Status: -
- Starting (White)
- Recovering (Yellow)
- Connecting (Orange)
- Connected (Green)
- Closing (Grey)

Order Client Status: - (if configured)
- Listening (Orange) - accepting connections
- Connected (Green) – one or more clients connected
- Stopping (Grey)

Trade Client Status: - (if configured)
- Listening (Orange) - accepting connections
- Connected (Green) – one or more clients connected
- Stopping (Grey)

SMARTS Client Status: - (if configured)
- Listening (Orange) - accepting connections
- Connected (Green) – one or more clients connected
- Stopping (Grey)

Order File Status: - (if configured)
- Open (Green)
- Closed (Grey)
- Error (Red)

Trades File Status: - (if configured)
- Open (Green)
- Closed (Grey)
- Error (Red)

SMARTS File Status: - (if configured)
- Open (Green)
- Closed (Grey)
- Error (Red)

2. Daily Cycle

CbJOtsET has been developed to run for multiple days. The application shuts down and wakes up at scheduled times throughout the day. This is configured within the CbJOtsET.ini file.

Refer: - 4.7 Daily Cycle Parameters

WAKE_TIME = time when program wakes up each morning,
SHUT_TIME = time when the program shutdown (hibernation) occurs.

Note: Currently there is no way of detecting Market Closed in the Cboe feeds so a timed shut-down and wake-up option has been created.

3. Installation

CbJOtsET should be installed as follows: -

<Install Directory>: - CbJOtsET.exe, CbJOtsET.ini
<Install Directory>/logs: - make a subdirectory for log files.
APP_LOG_DIRECTORY=logs.
<Install Directory>data: - make a subdirectory for data files.
APP_DATA_DIRECTORY=data

Prior to starting the application, the user is required to configure the CbJOtsET.ini file correctly. The CbJOtsET.exe file can then be executed to run the application.

The user must set the following parameters correctly: -

- Parameters – Fix Connection Configuration 4.1 FIX Connection Parameters
- Parameters – Fix Logon Configuration 4.2 FIX Logon Parameters

If the user requires the program to operate without a GUI screen, refer: - 4.4 Options Parameters

Note: When upgrading to a new version intra-day, the user should copy the FIX log file if installing in a new directory.

4 Configuration

Configuration parameters for the application are stored in the CbJOtsET .ini file.

4.1 FIX Connection Parameters

Primary: -
FIX_SERVER_HOST = Name of Cboe Fix Server.
E.g.: **FIX_SERVER_HOST**=CBOE_JP_FIX_ODROP

FIX_SERVER_PORT = Port to connect to for FIX.
E.g.: **FIX_SERVER_PORT**=10123

Cboe will supply values for these parameter settings.

4.2 FIX Session Parameters

FIX_SESSION_COUNT = The number of FIX Sessions to be created by the application, max 2.
E.g.: **FIX_SESSION_COUNT**=1

FIX_HEARTBEAT = Creates the heart beat intervals sent to the FIX server (seconds).
E.g.: **FIX_HEARTBEAT**=30

FIX_SENDER_ID = Part of Fix header, a valid value must be specified. Cboe will supply. (FIX_USER_ID for Cboe is currently treated the same)
E.g.: **FIX_SENDER_ID**=CbJ

FIX_SENDER_SUB_ID = Part of Fix header, a valid value must be specified. Cboe will supply.
E.g.: **FIX_SENDER_SUB_ID**=0001

FIX_TARGET_ID = Part of Fix header, a valid value must be specified. Cboe will supply.
E.g.: **FIX_TARGET_ID**=CbJ

FIX_TARGET_SUB_ID = Part of Fix header, a valid value must be specified. Cboe will supply.
E.g.: **FIX_TARGET_SUB_ID**=TEST

Note: RJE is not aware of any function of automatic password changing in Cboe order feeds.

4.3 Options Parameters

ENABLE_GUI = Indicates whether to run the application as a GUI.
E.g.: **ENABLE_GUI**=YES

ENABLE_PERF_STATS = Indicates whether to output the performance stats in the application log.
E.g.: **ENABLE_PERF_STATS**=YES

RECOVER_LOGS_AT_START = Indicates whether to recover the FIX during a restart.
E.g.: **RECOVER_LOGS_AT_START**=YES

RUN_ON_WEEKEND = Indicate whether to connect and logon during the weekend.
E.g.: **RUN_ON_WEEKEND**=NO

4.4 Functions Parameters

The only function offered by this application is:

1. Order Feed (& File)
2. Trade Feed (& File)
3. SMARTS Feed (& File)

The following parameters are found in the CbJOtsET.ini file and by setting them TRUE, the user can enable the above function.

Note: The user should set at least one of these functions TRUE in order to run the application.

4.4.1 Order Feed (and File)

ORDER_FEED=Indicates whether there is An order feed to facilitate order clients. The trades will output to a file in addition.

When the **ORDER_FEED** is set to TRUE, the application will support the order clients via a feed. The application will also produce an order file.

ORDER_FEED=TRUE

This also requires the following mandatory port setting. If the required port setting is not found in the CbJOtsET.ini file, the user cannot run the application until the correct settings are inputted.

ORDER_FEED_PORT=12009

4.4.2 Trade Feed (and File)

TRADE_FEED=Indicates whether there is a trade feed to facilitate trade clients. The trades will output to a file in addition.

When the **TRADE_FEED** is set to TRUE, the application will support the trade clients via a feed. The application will also produce a trade file.

TRADE _FEED=TRUE

This also requires the following mandatory port setting. If the required port setting is not found in the CbJOtsET.ini file, the user cannot run the application until the correct settings are inputted.

TRADE _FEED_PORT=41333

4.4.3 SMARTS Feed (and File)

SMARTS_FEED=Indicates whether there is a SMARTS feed to facilitate SMARTS clients. The SMARTS will output to a file in addition.

When the **SMARTS _FEED** is set to TRUE, the application will support the SMARTS clients via a feed. The application will also produce a SMARTS file.

SMARTS _FEED=TRUE

This also requires the following mandatory port setting. If the required port setting is not found in the CbJOtsET.ini file, the user cannot run the application until the correct settings are inputted.

SMARTS _FEED_PORT=12111

4.5 Logging Parameters

The Application log and FIX log are text files that can be used for trouble shooting.

APP_LOG_FILE=File name base for Application log. A new log is created each time the application is run. The application log includes the current date and time.

E.g.: **APP_LOG_FILE**=CbJOtsET.LOG
The filename produced will be (CbJOtsET.20230926_150113.log)

APP_LOG_DIRECTORY=Directory where the application log is stored.
E.g.: **APP_LOG_DIRECTORY**=logs

APP_DATA_DIRECTORY=Directory where the Orders file is stored.
E.g.: **APP_DATA_DIRECTORY**=data

APP_DATA_DIRECTORY defaults to APP_LOG_DIRECTORY if not specified.

FIX_LOG_FILE = file base for FIX Message Log. The filename always includes the current date.
E.g.: **FIX_LOG_FILE**=CbJOtsET.FIX

The filename produced will be (CbJOtsET.FIX.fix.messages.20230926.log).

FIX_LOG_DIRECTORY=Directory where FIX Message Log is stored.
E.g.: **FIX_LOG_DIRECTORY**=logs

If these settings are not specified, defaults settings will apply.

4.6 Daily Cycle Parameters

Refer: - 2. Daily Cycle

WAKE_TIME = time when program wakes up each morning (hour: in), default 07:00.
E.g.: **WAKE_TIME**=07:30

SHUT_TIME = time when the program shuts down, hibernation occurs (hour:min), default 23:30.
E.g.: **SHUT_TIME**=21:00

4.7 Configuration File Example

```
*******************************
* APPLICATION DETAILS        *
*******************************
APP_NAME=CbJOtsET
APP_VERSION=[Rev 01.00] [RJE Systems P/L 2023]
*******************************
* OPTIONS                    *
*******************************
ENABLE_GUI=YES
ENABLE_PERF_STATS=YES
RECOVER_LOGS_AT_START=YES
ENABLE_TEST_ORDERS=YES
FIX_MESSAGE_CHECKS=TRUE
RUN_ON_WEEKEND=NO
*******************************
* FIX SESSION PARAMETERS     *
*******************************
FIX_SESSION_COUNT=1
FIX_HEARTBEAT=30
FIX_THROTTLE_INTERVAL=500
* FIX_SERVER_HOST=CBOE_GW1
* FIX_SERVER_PORT=10472
* FIX_SENDER_ID=AURJ
* FIX_SENDER_SUB_ID=0001
* FIX_TARGET_ID=CXA
* FIX_TARGET_SUB_ID=TEST
* FIX_HEARTBEAT=5
*******************************
* FUNCTIONS                  *
*******************************
ORDER_FEED=TRUE
ORDER_VIEW=TRUE
ORDER_CANCEL=FALSE
SMARTS_FEED=TRUE
TRADE_FEED=TRUE
*******************************
* CLIENT PORTS               *
*******************************
ORDER_FEED_PORT=12009
```

```
*ORDER_CANCEL_PORT=12010
TRADE_FEED_PORT=41333
SMARTS_FEED_PORT=12111
*******************************
* MAXIMUM VALUES             *
*******************************
MAX_FEED_CLIENTS=32
*MAX_CANCEL_CLIENTS=120
MAX_ORDER_CLIENTS=32
MAX_CLORD_ID=9999999
MAX_TRADE_CLIENTS=32
MAX_SMARTS_CLIENTS=32
*******************************
* APP LOG FILE PARAMETERS     *
*******************************
APP_LOG_FILE=RJECboeJapan.LOG
APP_LOG_DIRECTORY=logs
APP_LOG_LEVEL=9
*******************************
* FIX LOG FILE PARAMETERS     *
*******************************
FIX_LOG_FILE=CbJOtsET.FIX
FIX_LOG_DIRECTORY=logs
*******************************
* DATA FILE PARAMETERS        *
*******************************
APP_DATA_DIRECTORY=data
APP_ORDERS_DATA_FILE=""
APP_SMARTS_DATA_FILE=""
APP_TRADES_DATA_FILE=""
*******************************
* PARTICIPANT IDENTITIES      *
*******************************
BROKER_ID=5892
CLEARING_FIRM=05512
TRADING_FIRM=RJES
*******************************
* WAKE/SHUT TIMES             *
*******************************
WAKE_TIME=08:00
SHUT_TIME=18:30
*******************************
* DATABASE PARAMETERS         *
*******************************
DATABASE_NAME=CboeJap
DATABASE_SERVER=RJEWKS2
DATABASE_PORT=5432
DATABASE_USER_ID=postgres
DATABASE_PASSWORD=rjeadmin
************ END **************
```

5. Failover and Disaster Recovery

The Cboe trading platform is based on a high availability, fault tolerant design, deployed across two sites in separate physical locations (Primary and Secondary) under an asynchronous configuration. The platform consists of two sets of the system in the primary site providing local redundancy and a backup system in the secondary site for failover. The design can be summarised as follows:

Hot Fail Over = > Primary to 'hot' Primary Stand by (@ Primary Site)
Warm Fail Over => Primary to 'warm' Secondary Stand by (@Secondary Site)

This application comes with a facility where it can be enabled to support Hot and Warm failovers.

5.1 Hot Failover

RJE recommends to have the hot failover option enabled in the application. This is to ensure that if there is an issue in the Primary Server, the application will make a seamless transition to the Primary Standby Server.

ENABLE_HOT_FAILOVER=YES

Setting details of the Standby Server are required to be provided in the CbJOtsET.ini file. Setting "ENABLE_HOT_FAILOVER=YES" without setting the details of the Standby Server, will prevent the application from starting.

FIX_STANDBY_HOST=CBOE_FD_SEC
FIX_STANDBY_PORT=10525

There are two additional settings of importance when attempting to connect to the Standby Server that are: -

FIX_CONNECT_TRY=2
FIX_CONNECT_INT=5

These settings determine how many times the application will make to connect to one server before attempting the next and how long it will wait in between attempts. This setting is not mandatory and if not set, the application will try both servers alternately without any wait time between the attempts.

5.2 Warm Failover

In the event of a total primary site failure, Cboe may failover to the secondary site that is maintained asynchronously. In order to maintain market integrity and enable participants to carefully manage their failover to the Cboe backup site, Cboe will perform a number of detailed verification steps to facilitate the participant failover and reconcile executed trade status with each participant.

Should such a failover be required, Cboe will communicate with both trading participants and vendors throughout the process.

During the failover process all open orders from the order book are purged (cancelled) and the order book is reset to an empty state. The status of trades executed by participants will be verified by Cboe operations staff.

In this instance, the following should be set in the CbJOtsET .ini file and the application restarted.

ENABLE_WARM_FAILOVER=YES

Setting details of the Secondary Server are required to be provided in the CbJOtsET .ini file. Setting "ENABLE_WARM_FAILOVER=YES" without setting the details of the Secondary Server, will prevent the application from starting.

FIX_STANDBY_HOST=CBOE_FD_SEC
FIX_STANDBY_PORT=10525

In some case, Cboe might want the client's sequence numbers to be incremented. This can be done in the following CbATradesET.ini settings.

FIX_SEQ_NO_OUT=193
FIX_SEQ_NO_IN=0

Important Note: Proper care should be taken before applying the above settings related to sequence numbers.

Once the system verification and reset processes have been completed by Cboe, and participant connectivity and readiness to connect to the backup market are verified, Cboe will then reopen the market and allow trading to recommence.

6 Data Output Options

There are 2 options for receiving the Cboe OTS data output from CbJOtsET.

1. TCP/IP Trades Data Feed
2. Orders File

6.1 TCP/IP Cboe Data Feed

One option for receiving orders, trades or SMARTS data is to make a TCP/IP connection to Cboe Japan orders data feed port and receive trades data in comma-delimited format. Data is simply sent when it is available; there is no need to request data.

In this case, all orders are single sided, and all data received from Cboe is included. Information on configuring the ports for client connections are in: - 4.5 Function Parameters

6.1.1 Order Feed (and File)

Most applications would process the header as it gives a list of field names corresponding to field positions. An example of the header output is below:

Country|S,Exchange|S,Market|S,FirmID|S,TraderID|S,ClientID|S,MsgSeqNo|N,OrderID|S,SecondaryOrderID|S,ClOrderID|S,OrigClOrdID|S,ExecID|S,TransactID|S,ExecRefID|S,ExecInst|S,ExecType|N,ExecBroker|N,Account|S,ClearingFirm|S,ClearingAccount|S,ClearingCrossRef|S,Symbol|S,Side|S,Price|N,OrderQty|N,OrderType|N,OrderCapacity|S,OrderStatus|N,ExecTransType|N,AvgPrice|N,LastMarket|S,LastPrice|N,LastShares|N,LastCapacity|N,CumQty|N,LeavesQty|N,TimeInForce|N,ExpireTime|T,MinQty|N,MaxFloor|N,TradeLiqIndicator|S,ExecRestatementReason|S,Undisclosed|S,ShortSellNakedQty|S,ShortSellCoveredQty|S,ShortSellLongQty|S,NoSelfTrade|S,NoSelfTradeOrderNum|S,RemoveCrossFromClearing|S,IDSource|S,SecurityID|S,SecurityExchange|S,PriceImprovement|N,MidPointExecution|C,ExecVenue|S,NoTradeFeat|C,OriginOfTrans|S,IntermediaryID|S,DirectedWholesaleIndex|C,OrderRestriction|C,OnMarketCrossType|S,OnMarketTradeReportType|C,ContraTradePA|S,ContraOriginOfTrans|S,ContraIntermediaryID|S,ContraDirectedWholesale|S,TransactTime(UTC)|T,ExecutedDate(Local)|D,ExecutedTime(Local)|T,~

6.1.2 Trades Feed (and File)

Most applications would process the header as it gives a list of field names corresponding to field positions. An example of the header output is below:

Country|S,Exchange|S,Market|S,FirmID|S,TraderID|S,ClientID|S,MsgSeqNo|N,OrderID|S,ClOrderID|S,ExecID|S,ExecRefID|S,ExecType|N,ExecBroker|N,Account|S,Symbol|S,Side|S,OrderQty|N,OrderType|N,OrderStatus|N,ExecTransType|N,AvgPrice|N,LastPrice|N,LastShares|N,LastCapacity|N,TradeValue|N,CumQty|N,LeavesQty|N,TradeLiqIndicator|S,TransactTime(UTC)|T,ExecutedDate(Local)|D,ExecutedTime(Local)|T,TimeStamp(UTC)|TS,VWAPSessionID|S,~

6.1.3 SMARTS Feed (and File)

Most applications would process the header as it gives a list of field names corresponding to field positions. An example of the header output is below:

Country|S,Exchange|S,Market|S,FirmID|S,TraderID|S,ClientID|S,MsgSeqNo|N,OrderID|S,ClOrderID|S,OrigClOrdID|S,ExecID|S,ExecRefID|S,ExecInst|S,ExecType|N,ExecBroker|N,Account|S,Symbol|S,Side|S,Price|N,OrderQty|N,OrderType|N,OrderCapacity|S,OrderStatus|N,ExecTransType|N,Description|S,AvgPrice|N,LastPrice|N,LastShares|N,LastCapacity|N,CumQty|N,LeavesQty|N,TimeInForce|N,ExpireTime|T,MaxFloor|N,PegDifference|N,PostOnly|S,TradeLiqIndicator|S,ExecRestatementReason|S,TransactTime(UTC)|T,ExecutedDate(Local)|D,ExecutedTime(Local)|T,TimeStamp(UTC)|TS,Pullback|S,CHIXSessionID|S,HFTflag|C,~

6.1.4 Orders Comma-Delimited Data

The same format is used for all trades. Fields that are not relevant are simply empty. An example of the data output is shown below: -

JAPAN,CBOE-J,CBOE-J,CXJS,TEST,,,17S9ZD98OD2U,,1318000,,S10000NKX3,!!!!!!,,,0,,,!!!!!!,!!!!!!,!!!!!!,2042,1,7716.00,500,2,A,0,0,0,,0,0,,0,500,3,,,,R,,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,3,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,N,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,!!!!!!,20230921-03:18:10.041275,10101,0:00:00,~

6.1.5 Trades Comma-Delimited Data

The same format is used for all trades. Fields that are not relevant are simply empty. An example of the data output is shown below: -

JAPAN,CBOE-J,CBOE-J,CXJS,TEST,,0,17TG2NZYGQA3,000000005S8000001216,0100007CK,,1,,,2043,2,200,2,1,0,22008.00,22008.00,100,3,2200800.00,100,100,A,20231017-23:20:14.125284,20231018,10:20:14,20231017-23:20:14,,~

6.1.6 SMARTS Comma-Delimited Data

The same format is used for all trades. Fields that are not relevant are simply empty. An example of the data output is shown below: -

JAPAN,CBOE-J,CBOE-J,CXJS,TEST,,,FCTST0008P13I0000001,CTST00085S8000000001,,FCTST0008P13I0000001,,8,,,2041,2,4108.00,200,2,,8,0,A: Outside trading hours,0,0,0,,0,0,4,,,,,,20231017-23:00:00.084062,20231018,10:00:00,20231017-23:00:00,,,,~

6.2 Orders File

A .orders data file is produced each day with, or without the comma-delimited header. A separate line is created for each trade record. The contents of the trades file are identical to the data that is sent in the trades feed.

On a restart the internal copy of the trades is recreated from the FIX Message Log.

Configuration settings for trades data outputs are: -

APP_DATA_DIRECTORY=Directory where the Orders file is stored.
APP_DATA_DIRECTORY defaults to APP_LOG_DIRECTORY if not specified.

7 FIX Session Sequence Numbers

Typically FIX session sequence numbers start from 1 each day.

By default, when reconnecting/restarting the sequence numbers at both ends continue on from their previous values and any missing messages are recovered.

This is true for Cboe as the version of this they are using, 4.2, does not support the resetting of sequence numbers.

Therefore, on a restart, the application reprocesses the FIX Message log to reestablish from/to sequence numbers.

7.1 FIX Message Log

Typically, the FIX session is continued across runs and there is a single FIX Message log for each day. In that mode, traffic sent / received (including trades) is recovered from the FIX Messages log at startup.

Therefore, all orders data is rebuilt from the message log each start up.

When resuming the FIX session, the application typically only fetches new orders.

You can specify a filename/ directory for this file in: - 4.6 Logging Parameters:

The FIX message log should never be deleted from CbJOtsET. If in the rare event the FIX message log is corrupted, the user should rename the file.

7.2 Missing FIX Message Log

A missing FIX message log could be cause by the following: -

- Running from a different directory or with different CbJOtsET.ini settings.
- Deleting or renaming the file.

For CbJOtsET, this can cause problem with the sequence number of the login message we send to Cboe. If the sequence number is less than expected Cboe will ignore this message and you will eventually get the following error: -

```
*********************************************************************************************
*** Fatal Error - Exceeded FIX Logon retries - Check Config FIX_SERVER_PORT / FIX_SERVER_HOST.
***
*** If settings above are correct then could be a problem with the Fix Message log.                    ***
*** See CbJOtsET Rev 1.0.pdf Missing Fix Message Log.                                                  ***
*** You can run CbJOtsET -s 'nnn'.   Where 'nnn' = last message sequence no from our end.              ***
*********************************************************************************************
```

This error could mean the FIX Message log has been deleted or you could be connecting to the wrong host / port.

Note: A message log error can only be a problem if user has previously connected successfully.

7.3 Specifying a Restart Sequence No

If the user knows what the outbound FIX sequence number from their end should be, the user can specify the following: -

- CbJOtsET –so nnn – Where nnn is the sequence number.

You should be able to get the sequence number from the previous FIX Message Log.

If you don't know the sequence number you can obtain it from Cboe or Cboe can reset the FIX session.

In this mode the application will re-request all trades for the day from Cboe.

If you specify an incorrect sequence number the application will still attempt to recover, but it is strongly recommended you restart with the correct number.

8. Field Mappings CbJOtsET <-> FIX

The following is the Field Mapping for:

- Orders
- Trades
- SMARTS

8.1 Orders Field Mapping

Orders Field Name	FIX Field Name	FIX Field No
country	RJE	"JAPAN"
exchange	RJE	"CBOE - J"
Market	RJE	"CBOE - J"
account	Account	1
avg_price	AvgPx	6
cl_order_id	ClOrdID	11
cum_qty	CumQty	14
exec_id	ExecID	17
exec_ref_id	ExecRefID	19
exec_trans_type	ExecTransType	20
id_source	IDSource	22
last_capacity	LastCapacity	29
last_price	LastPx	31
last_shares	LastShares	32
message_no	MsgSeqNo	34
order_qty	OrderQty	38
order_status	OrdStatus	39
price	Price	44
order_capacity	OrderCapacity	47
firm_id	SenderCompID	49
trader_id	SenderSubID	50
side	Side	54
symbol	Symbol	55
time_in_force	TimeInForce	59
transact_time	TransactTime(UTC)	60
exec_broker	ExecBroker	76
client_id	ClientID	109
min_qty	MinQty	110
max_floor	MaxFloor	111
expire_time	ExpireTime	126
exec_type	ExecType	150
leaves_qty	LeavesQty	151

exec_restatement_reason	ExecRestatementReason	378
cash_margin	CashMargin	544
risk_group_id	RiskGroupID	7699
no_trade_feat	NoTradeFeat	7713
no_trade_key	NoTradeKey	7714
VWAPSessionID	VWAPSessionID	8004
ord_class	OrderClassification	8060
pull_back	PullBack	8105
mod_seq	ModifySequence	9617
orig_comp_id	OrigCompID	9688
orig_sub_id	OrigSubID	9689
working_price	WorkingPrice	9690
trade_liquidity_id	TradeLiquidityIndicator	9730
fee_code	FeeCode	9882
order_type	OrderType	
last_market	LastMarket	
prev_trd_px	PrevTrdPx	
prev_trd_qty	PrevTrdQty	
prev_trd_liq_ind	PrevTrdLiqInd	
tost_net_ord_id	TostNetOrdID	
tost_net_exc_id	TostNetExeID	
tost_net_trans_time	TostNetTransTime	
marg_trans_type	MargTransType	
ord_text	OrderText	
rout_inst	RoutInst	

8.2 Trades Field Mapping

CbJTradesET Field Name	**FIX Field Name**	**FIX Field No**
Country	RJE	"JAPAN"
Exchange	RJE	"CBOE - J"
Market	RJE	"CBOE - J"
TradeValue	RJE	Calculated
Account	Account	1
AvgPrice	AvgPx	6
clOrderId	ClOrdID	11
CumQty	CumQty	14
exec_id	ExecID	17
execRefID	ExecRefID	19
execTransType	ExecTransType	20
LastCapacity	LastCapacity	29
LastPrice	LastPx	31
quantity	LastShares	32
trade_no	MsgSeqNo	34

orderId	OrderID	37
OrderQty	OrderQty	38
OrdStatus	OrdStatus	39
OrdType	OrdType	40
firm_id	SenderCompID	49
Side	Side	54
symbol	Symbol	55
trader_id	TargetCompID	56
TransactTime	TransactTime	60
trade_date	From TransactTime localtime	60
time_trade	TransactTime localtime	60
utc_timestamp	TransactTime utctime	60
ExecBroker	ExecBroker	76
ClientID	ClientID	109
ExecType	ExecType	150
LeavesQty	LeavesQty	151
VWAPSessionID	VWAPSessionID	8004
TradeLiquidId	TradeLiquidityIndicator	9730

8.3 SMARTS Field Mapping

SMARTS Field Name	**FIX Field Name**	**FIX Field No**
country	(Internal)	-
exchange	(Internal)	-
market	(Internal)	-
order_active	(Derived)	-
account	Account	1
avg_price	AvgPx	6
clord_id	ClOrdID	11
cum_qty	CumQty	14
exec_id	ExecID	17
exec_trans_type	ExecTransType	20
last_price	LastPx	31
last_fill	LastShares	32
message_no	MessageSeqNo	34
order_id	OrderID	37

order_qty	OrderQty	38
order_status	OrdStatus	39
order_type	OrdType	40
price	Price	44
order_capacity	Capacity	47
firm_id	SenderCompID	49
side	Side	54
symbol	Symbol	55
trader_id	TargetCompID	56
description	Text	58
time_in_force	TimeInForce	59
transact_time	TransactTime	60
trade_date	TradeDate	75
client_id	ClientID	109
min_qty	MinQty	110
max_floor	MaxFloor	111
exec_type	ExecType	150
leaves_qty	LeavesQty	151
no_of_fills	(Derived)	-
time_stamp	(Database Update Time)	-