

Revision:	2
1. Overview	3
1.1 Features:	4
1.2 GUI Screen:	5
2. Daily Cycle	7
3. Installation	7
4. Configuration	8
4.1 COG Connection Parameters:	8
4.2 COG Logon Parameters:	8
4.3 Optional COG Parameters:	9
4.4 Order Feed Parameters:	9
4.5 Logging Parameters:	10
4.6 Daily Cycle Parameters:	10
4.7 Feature Filter Parameters:	11
4.8 Other Parameters:	11
4.9 Configuration File Example :	12
5. Comma-Delimited Application Development	13
5.1 Comma-Delimited Header:	13
5.2 Comma-Delimited Data:	14
5.3 Orders File:	14
6. Message Sequence Numbers	15
6.1 FIX Message Log:	15
6.2 Missing FIX Message Log:	15
6.3 Specifying a Restart Sequence No:	16
7. Features for Testing	17
7.1 Stop and Start:	17
7.2 User set Message Sequence Numbers:	17
8. Command Clients	18
8.1 Order Cancellation:	18
8.2 Supported Cancellation Types:	18
8.3 Command Clients Parameters:	19
8.4 Cancellation Database Updates:	19
9. Database	19
9.1 Database Tables:	20
9.2 Database Parameters:	27
9.3 npgsql files:	27
9.4 SQL Script Files:	28
Appendix 1	29
Order Feed:	29
Order Table:	31

This Document:

MCOrdersCHXA[COG].pdf Rev 1.1

This document details how to install, configure and run MCOrdersCHXA[COG]

Revision:

05/12/2012 – Rev 01.00 - vaasuGI – Produced the first version of this manual.

28/11/2013 – Rev 01.10 – vaasuGI – Included the newly added fields in Order Feed and Order Table and updated the relevant DB scripts.



1. Overview

MCOOrdersCHXA[COG] application communicates with the Chi-X Australia Cross Session Order Management Gateway (COG) via FIX protocol. It extracts Orders from the consolidated feed of FIX messages supplied by the COG service. It then provides the Orders via comma-delimited feeds as well as stores them in the Database. It also allows external clients to connect and routes their Order Cancel Requests to the Gateway.

The following diagram depicts the overall functionality and connectivity of the MCOOrdersCHXA[COG] production system.

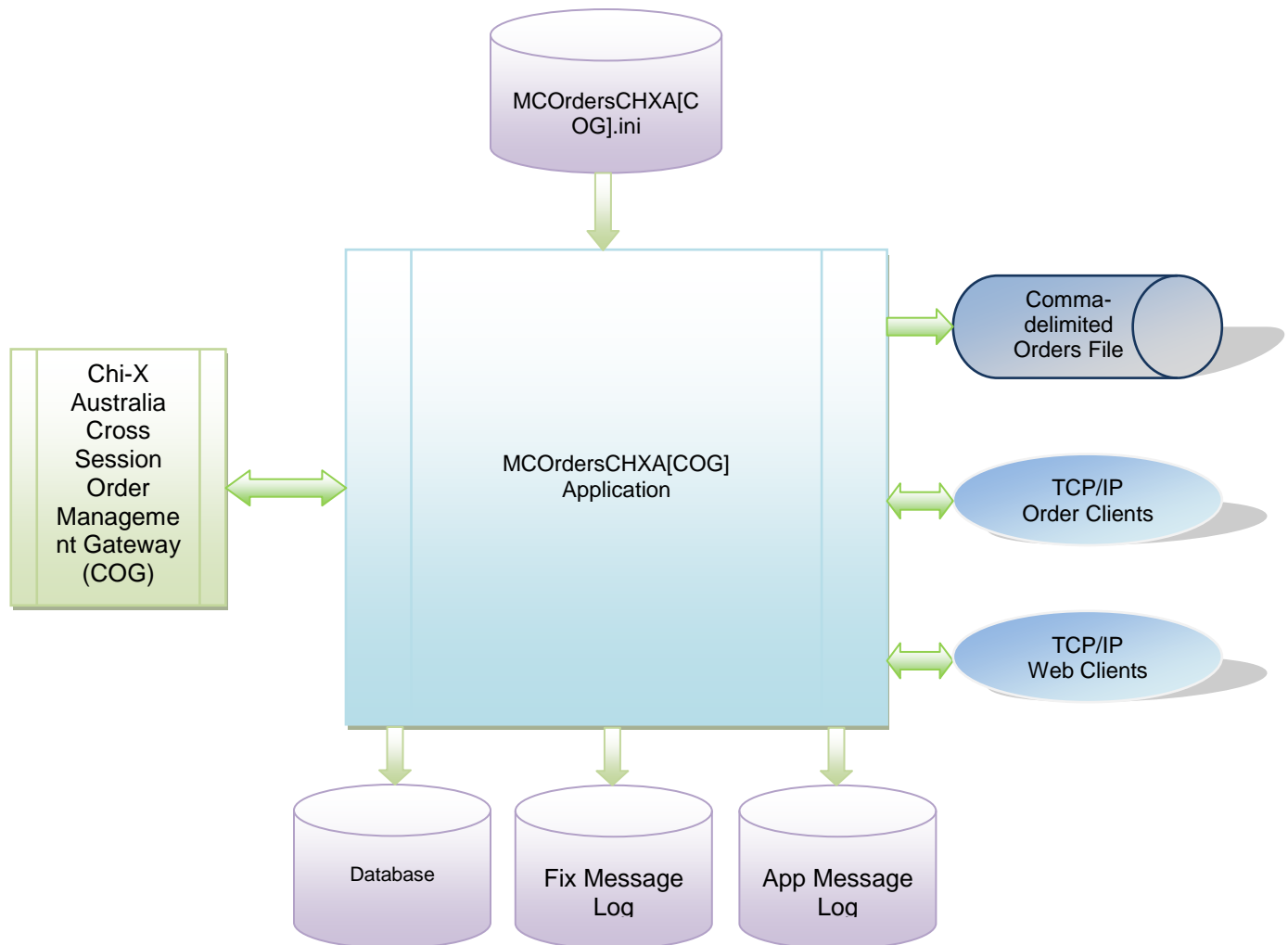


Figure 1. The MCOOrdersCHXA[COG] Production System



1.1 Features:

Order Feed

Orders are available in the following output forms. Order feed consists of all the Execution Reports extracted from Chi-X Australia Trading System.

- Comma-delimited order file (single trade side)
- Comma-Delimited TCP/IP order feed (single trade side)

Note: The Comma-Delimited TCP/IP feed is similar to all other MCTrades products.

Data Store

Orders extracted from Chi-X Australia Trading System are stored in the database table `chxa_orders`. A database is required for this feature. More details can be found in [9. Database](#)

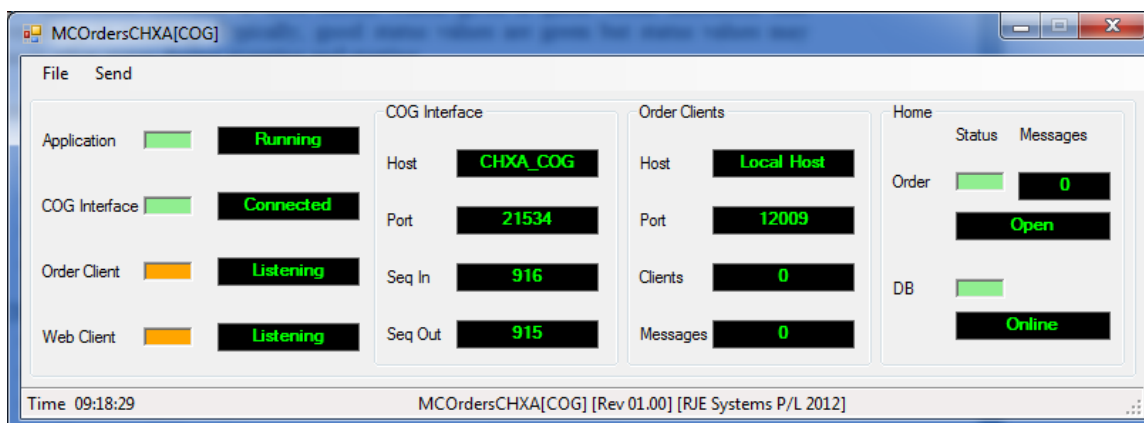
Order Cancellation

An external GUI client or a PHP based web client can connect and route the cancel requests via MCOOrdersCHXA[COG].

More details can be found in [8. Command Clients](#)

The user can configure the application to enable/disable any of the above features. More details can be found in [4.7 Feature Filter Parameters:](#)

1.2 GUI Screen:



The application contains a GUI screen which gives a quick visual indication that everything is working. Typically, good status values are green but status values may transit other states during stopping and starting.

Application Status:-

- Starting (Orange)
- Running (Green) – normal
- Stopping (Red)
- Hibernating (Grey) – normal overnight.
- Waiting (Grey) – normal when user press 'stop'

COG Interface Status:-

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

Order Clients Status: - (if accepting client connections)

- Listening (Orange) - accepting connections
- Connected (Green) – one or more clients connected
- Stopping (Grey)

Web Clients Status: - (if accepting client connections)

- Listening (Orange) - accepting connections
- Connected (Green) – one or more clients connected
- Stopping (Grey)

Orders File Status:-

- Open (Green)
- Closed (Grey)
- Error (Red)

Database Status:-

- Open (Green)
- Not connected (Grey)



2. Daily Cycle

MCOOrdersCHXA[COG] can be run for multiple days; it shuts down and wakes up at a certain scheduled time each day.

Refer [4.6 Daily Cycle Parameters](#):

Note: We currently have no way of detecting Market Close in Chi-X Australia Cross Order Management service. A timed shutdown is the only option.

3. Installation

Install MCOOrdersCHXA[COG] as follows :-

<Install Directory>:- MCOOrdersCHXA.exe, MCOOrdersCHXA[COG].ini

<Install Directory>:- Mono.Security.dll, Npgsql.dll

<Install Directory>/logs: - Make a subdirectory for logs, trades and orders files.

To run the program, run MCOOrdersCHXA.exe, with the presence of a correctly configured MCOOrdersCHXA[COG].ini file and the two dll files Mono.Security.dll and Npgsql.dll.

You must set the following parameters correctly:-

- Parameters – COG Connection Configuration [4.1 COG Connection Parameters](#):
- Parameters – COG Logon Configuration [4.2 COG Logon Parameters](#):

If you wish to run the program without a GUI, refer [4.8 Other parameters](#):

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

4 Configuration

All configuration parameters are stored in MCOOrdersCHXA[COG].ini

4.1 COG Connection Parameters:

COG_SERVER_HOST = Name of Chi-X Australia Cross Order Management Gateway
e.g **COG_SERVER_HOST**=CHXA_COG

COG_SERVER_PORT = Port to connect to Chi-X Australia Cross Order Management Gateway
e.g **FIX_SERVER_PORT**=21534

Chi-X Australia will supply values for these parameter settings.

Note: IP address of the Cross Order Management Gateway host can be entered in the ini file. Alternatively an entry for the COG host can be made in the windows host file (C:\Windows\System32\drivers\etc\hosts) with a name for example CHXA_COG and that name in turn can be used in the ini file.

4.2 COG Logon Parameters:

COG_SENDER_ID = Part of Fix header, a valid value must be specified.
e.g **COG_SENDER_ID**=COGRJE01

COG_TARGET_ID = Part of Fix header, a valid value must be specified.
e.g **COG_TARGET_ID**= CXA

Chi-X Australia will supply values for these parameter settings.

4.3 Optional COG Parameters:

COG_HEARTBEAT=<Heartbeat interval> (Seconds) – default = 5

Example **COG_HEARTBEAT**=5

Note: You should consult Chi-X Australia before setting this parameter, the default of 5 seconds is recommended.

4.4 Order Feed Parameters:

This is the TCP/IP port that applications can connect to receive a feed of execution report data.

The format of the data is described in [5. Comma Delimited Application Development](#)

ORDERS_PORT = TCP/IP port for all Orders.

e.g. **ORDERS_PORT**=12009

4.5 Logging Parameters:

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

APP_LOG_FILE = file base for application log, a new log is taken each run; the application log includes the current date and time.

e.g **APP_LOG_FILE**= MCOOrdersCHXA[COG]

The name of the file e.g MCOOrdersCHXA[COG].App.Messages.20120808_093415.log

FIX_LOG_FILE = file base for FIX Message Log; the filename always includes the current date.

e.g **FIX_LOG_FILE**= MCOOrdersCHXA[COG]

The name of the file e.g MCOOrdersCHXA[COG].Fix.Messages.20120423.log

APP_LOG_DIRECTORY=Directory where the application log is stored.

e.g **APP_LOG_DIRECTORY**=logs

FIX_LOG_DIRECTORY=Directory where FIX message log is stored.

e.g **FIX_LOG_DIRECTORY**=logs

APP_DATA_DIRECTORY=Directory where output files are stored.

e.g **APP_DATA_DIRECTORY**=data

Note: APP_DATA_DIRECTORY defaults to APP_LOG_DIRECTORY if not specified.

If you don't specify these settings, defaults will apply.

Note: In this application the FIX Message Log is important see [6.1 FIX Message Log](#): for more details.

4.6 Daily Cycle Parameters:

Refer [2. Daily Cycle](#)

WAKE_TIME = time when program wakes up each morning (hour:min), default 08:00.

e.g **WAKE_TIME**=08:00

SHUT_TIME = time when program shuts down each day (hour:min) default 16:00.

e.g **SHUT_TIME**=16:00

4.7 Feature Filter Parameters:

Refer [1.1 Features:](#)

ORDER_FEED=TRUE – set TRUE to enable Order Feed

ORDER_CANCEL=TRUE – set TRUE to enable Order Cancel from Command Clients

DATABASE=TRUE – set TRUE to enable database connection and storing of orders in the database

Note: Enabling Order Cancel, automatically enables DATABASE as well, as data store is required for Order Cancellation.

4.8 Other Parameters:

NO_GUI=YES – set to enable the application running with no GUI

See also

- [8.3 Command Clients Parameters:](#)
- [9.2 Database Parameters:](#)



4.9 Configuration File Example :

```

*****
* APPLICATION DETAILS *
*****
APP_NAME=MCOOrdersCHXA[COG]
INI_VERSION=1.0.2.0
APP_VERSION_POSTFIX=RJE Systems P/L 2013
*****
* WITH/WITHOUT GUI *
*****
*NO_GUI=YES
*****
* COG SESSION PARAMETERS *
*****
COG_SERVER_HOST=CHXA_COG
COG_SERVER_PORT=21534
COG_SENDER_ID=COGRJE01
COG_TARGET_ID=CXA
COG_HEARTBEAT=5
*****
* TCP CLIENTS' PARAMETERS *
*****
ORDERS_PORT=12009
COMMAND_PORT=12010
*****
* APP LOG FILE PARAMETERS *
*****
APP_LOG_FILE=MCOOrdersCHXA[COG]
APP_LOG_DIRECTORY=logs
APP_LOG_LEVEL=9
*****
* FIX LOG FILE PARAMETERS *
*****
FIX_LOG_FILE=MCOOrdersCHXA[COG]
FIX_LOG_DIRECTORY=logs
*****
* PARTICIPANT IDENTITIES *
*****
BROKER_ID=ABN01
CLEARING_FIRM=11111
TRADING_FIRM=RJE
*****
* FEATURE FILTER *
*****
ORDER_FEED=TRUE
ORDER_CANCEL=TRUE
DATABASE=TRUE
*****
* MAXIMUM VALUES *

```

```

*****
MAX_CMD_CONNECTIONS=120
MAX_ORDER_CLIENTS=32
MAX_CLORD_ID=9999999
*****
* WAKE/SHUT TIMES *
*****
WAKE_TIME=08:00
SHUT_TIME=17:00
*****
* DATABASE PARAMETERS *
*****
DATABASE_NAME=webdb
DATABASE_SERVER=127.0.0.1
DATABASE_PORT=5432
DATABASE_USER_ID=postgres
DATABASE_PASSWORD=rjeadmin
***** END *****

```

5 Comma-Delimited Application Development

One option for developers is to make a TCP/IP separate connections to MCOOrdersCHXA[COG] order feed port and receive order data in comma-delimited format separately. Data is simply sent when it is available; there is no need to request data. All execution reports received from Chi-X Australia are included.

The port for clients' connections is configured in [4.4 Order Feed Parameters](#):

5.1 Comma-Delimited Header:

Most applications would process the header as it gives a list of field names corresponding to field positions.

Orders

```

Country|S,Exchange|S,Market|S,FirmID|S,TraderID|S,ClientID|S,MsgSeqNo|N,OrderID|
S,SecondaryOrderID|S,CiOrderID|S,OrigCiOrdID|S,ExecID|S,TransactID|S,ExecRefID|S
,ExecInst|S,ExecType|N,ExecBroker|N,Account|S,ClearingFirm|S,ClearingAccount|S,Cle
aringCrossRef|S,Symbol|S,Side|S,Price|N,OrderQty|N,OrderType|N,OrderCapacity|S,Ord
erStatus|N,ExecTransType|N,AvgPrice|N,LastPrice|N,LastShares|N,LastCapacity|N,Cum
Qty|N,LeavesQty|N,TimeInForce|N,ExpireTime|T,MinQty|N,MaxFloor|N,TradeLiqIndic
ator|S,ExecRestatementReason|S,Undisclosed|S,ShortSellNakedQty|S,ShortSellCovered

```

Qty|S,ShortSellLongQty|S,NoSelfTrade|S,NoSelfTradeOrderNum|S,RemoveCrossFromC
 learing|S,IDSsource|S,SecurityID|S,SecurityExchange|S,PriceImprovement|N,MidPointEx
 ecution|C,ExecVenue|S,NoTradeFeat|C,OriginOfTrans|S,IntermediaryID|S,DirectedWhol
 esaleIndex|C,OrderRestriction|C,OnMarketCrossType|S,OnMarketTradeReportType|C,Tr
 ansactTime(UTC)|T,ExecutedDate(Local)|D,ExecutedTime(Local)|T,TimeStamp(UTC)|
 TS,~

5.2 Comma-Delimited Data:

Fields that are not relevant are simply empty.

Orders

```
AUSTRALIA,chi-x AUS,chi-x
AUS,CXAPTE,COGRJE01,RJE03,13,62,,1100000,,10000000,,,,0,RJE,ABN01,1
1111,,,BHP,1,37.40,1000,2,A,0,0,0.00,0.00,0,,0,1000,0,,,,,,,,,,,,,,,,,,,,,2013112
8-00:00:06.975,20131128,11:00:06,20131128-00:00:06,~
```

Note: Additional examples are available from RJE.

5.3 Orders File:

An Order file is produced for each day with a comma-delimited header and a comma-delimited execution record for each execution. The contents of this file are identical to the data that would be sent of an orders feed.

On a restart mid-day, the internal copy of the orders is recreated from the FIX Message Log and the old orders file gets replaced by a new orders file.
 e.g MCOOrdersCHXA[COG].20120808.orders



6 Message Sequence Numbers

Message Sequence Numbers start from 1 each day. By default when reconnecting/restarting mid-day, the sequence numbers at both ends continue on from their previous values and any missing messages are recovered. Hence, on a restart the application reprocesses the FIX Message log to re-establish outbound/inbound sequence numbers. The version of FIX protocol being used (version 4.2) doesn't support resetting sequence numbers.

6.1 FIX Message Log:

Typically the FIX session is continued across runs and there is a single FIX Message log for each day. Messages sent/received are recovered from the FIX Messages log at startup. When resuming the FIX session the application only fetches the new messages.

You can specify a filename/directory for this file in [4.5 Logging Parameters](#):

Note: You should never delete the FIX message log, if the rare event that is corrupted, you should rename the file.

6.2 Missing FIX Message Log:

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

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

This can cause problems with the sequence number of the login message we send to the Chi-X Trading System. If the sequence number is less than expected Chi-X will ignore this message and the application will eventually close with the following error logged.

```
FATAL ERROR ~|[Session] @Logging onto CHI-X Cross Session Order Management Gateway. [ERROR] Exceeded logon retries.
```

```
FATAL ERROR ~|[Session] @Logging onto CHI-X Cross Session Order Management Gateway. [ERROR] Failed to logon.
```

```
FATAL ERROR ~|[DailyRun] Application stopping. [ERROR] Fatal Error occurred.
```

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

Note: A message log error can only be a problem if you have successfully connected earlier.

6.3 Specifying a Restart Sequence No:

If you know what the outbound FIX sequence number from your end should be, you can specify it as follows

MCOOrdersCHXA -S nnn

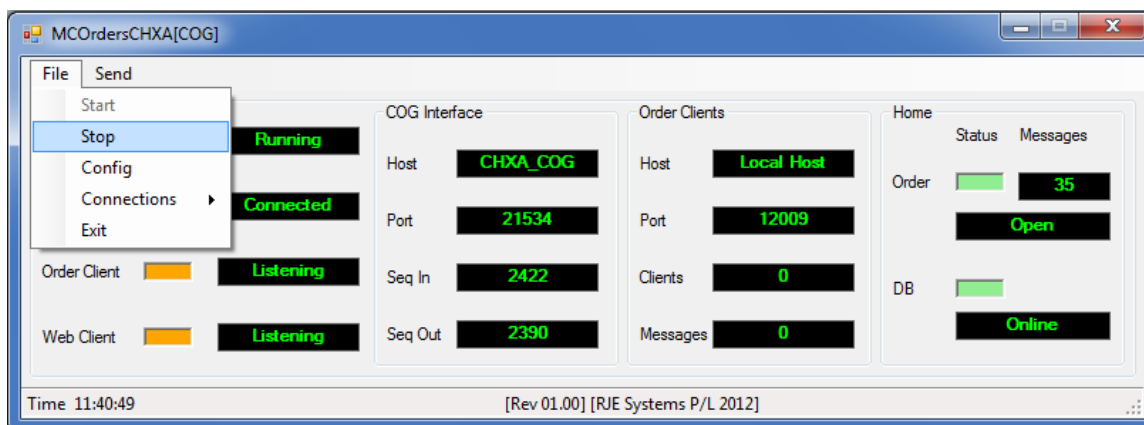
Note: Where nnn is the sequence number.

You should be able to get the number from the previous FIX Message Log. If you don't know this number you can obtain it from the Chi-X administrator or he can reset the FIX session (as the last option). In this mode the application will re-request all trades for the day from Chi-X.

7 Features for Testing

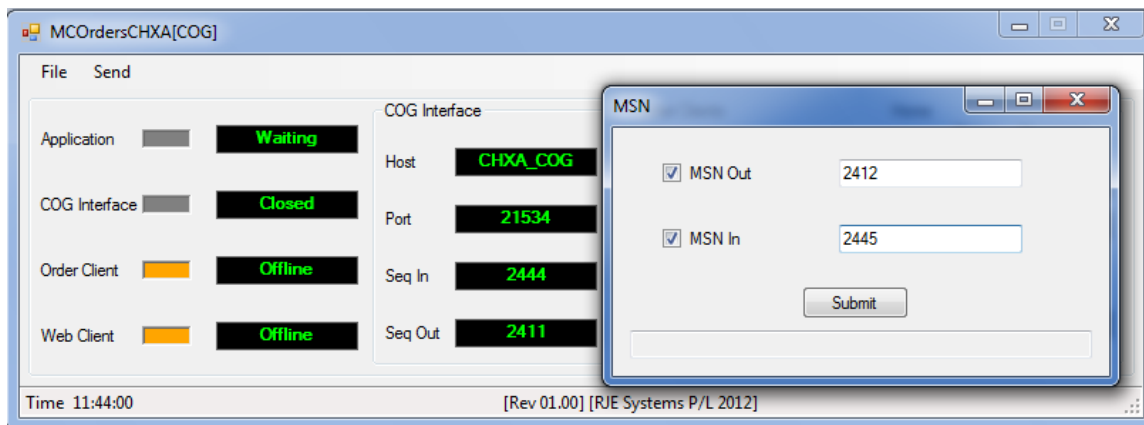
The following features are included solely for the purpose of facilitating the testing of this application.

7.1 Stop and Start:



The 'Stop' and the 'Start' submenus provided under the File menu facilitate stopping and restarting the application without exiting from the application.

7.2 User set Message Sequence Numbers:



When restarting the application via Stop/Start menu, the user is given the option of changing the last outbound (MSN Out) and inbound (MSN In) message sequence numbers in MCOOrdersCHXA[COG] application. This feature could be used during the conformance tests to see how both parties react (application and exchange) for such changes in the message sequence number.

8 Command Clients

8.1 Order Cancellation:

An external GUI client or a PHP based web client can connect and route the cancel requests via MCOOrdersCHXA[COG].

The following FIX messages are related to order cancellation

- Order Cancel Request
- Order Cancel Acknowledgement
- Order Cancel Reject

When an order cancel request succeeds an Order Cancel Acknowledgement message, which is one kind of Execution Report message, is sent by the exchange. An Order Cancel Reject is only sent when an order cancel request fails. MCOOrdersCHXA[COG] stores the cancel results in 'chxa_order_cancel_result' table in the database.

8.2 Supported Cancellation Types:

Only one type of order cancel is currently supported:-

1. Cancel Individual Order:-
CANCEL_REQUEST|USER=admin|REQUEST_NO=297|CxlType=F|OrderID=6688077|~

8.3 Command Clients Parameters:

COMMAND_PORT= TCP port command clients connect to.
COMMAND_PORT=12010

8.4 Cancellation Database Updates:

Table :- **chxa_last_clord_id** this table is used to ensure a unique ClOrdId for each cancellation transaction. It is updated after each cancellation request to ensure each request has a unique id.

Table :- **chxa_trans** web clients create an entry in this table each time they issue a cancellation request. MCOrdersCHXA[COG] updates this table when the cancellation request result is known. It is intended that web clients will archive the contents of this table.

Table :- **chxa_order_cancel_result** this table is updated with the results of each order cancel request. The intention is that this table will be a long term 'audit trail' of cancellation activity.

9 Database

Database design, tables and functions have been developed and tested with a PostgreSQL database running under Windows and Linux.

The MCOrdersCHXA[COG] application uses the "npgsql" .net data provider for PostgreSQL. It calls PostgreSQL Functions (Stored Procedures) for database access and updating.

9.1 Database Tables:

Table - system

The purpose of this table is to allocate a unique Guid (uuid) to each system. In this context MCOOrdersCHXA[COG] is one system. All data of MCOOrdersCHXA[COG] has the system_id of MCOOrdersCHXA.

Function :- `get_system_info()` create/retrieve system table information for a particular system.

```
-- Table: "system"

-- DROP TABLE "system";

CREATE TABLE "system"
(
    id bigserial NOT NULL,
    system_id uuid,
    system_name character varying(50),
    exchange character varying(10)
)
WITH (
    OIDS=TRUE
);
ALTER TABLE "system" OWNER TO postgres;
GRANT ALL ON TABLE "system" TO postgres;
GRANT ALL ON TABLE "system" TO public;
```

Table – system_state

This table shows the current state of a system indicating if the system is ready to store the data.

Currently defined system states are:-

```
enum SessionState : int
{
    Connecting = 10,
    Connected  = 20,
    Ready      = 30,
    Closed     = 90
}
```

The table is also updated periodically to provide `memory_trans` and `database_trans` counters. These provide feedback of whether `orders` table update is keeping with the rate execution reports are being sent by the Chi-X Australia Cross Order Management Gateway.

Function :- `update_system_state()` - updates the `system_state` table.

```
-- Table: system_state
-- DROP TABLE system_state;

CREATE TABLE system_state
(
    id bigserial NOT NULL,
    system_id uuid,
    system_state integer,
    host_name character varying(50),
    port_no integer,
    memory_trans integer,
    database_trans integer,
    last_update timestamp without time zone
)
WITH (
    OIDS=TRUE
);
ALTER TABLE system_state OWNER TO postgres;
GRANT ALL ON TABLE system_state TO postgres;
GRANT ALL ON TABLE system_state TO public;
```

Table – chxa_orders

This is the main table of interest which stores order data. As execution reports occur the current state of the database is updated to reflect the current state of the order. When the field order_active='Y' the order is an active order which is a candidate for cancellation. As orders trade out or are cancelled order_active is set to 'N'.

The orders information is kept in the DB indefinitely as it may be useful.

Function :- chxa_update_order() – Updates the orders table for each execution report

```
-- Table: chxa_orders

-- DROP TABLE chxa_orders;

CREATE TABLE chxa_orders
(
  id bigserial NOT NULL,
  system_id uuid NOT NULL,
  country character varying(10),
  exchange character varying(10),
  market character varying(10),
  firm_id character varying(30),
  trader_id character varying(30),
  client_id character varying(30),
  message_no integer,
  order_active character(1),
  order_id character varying(50) NOT NULL,
  secondary_order_id character varying(50),
  clord_id character varying(50),
  org_clord_id character varying(50),
  exec_id character varying(50),
  transact_id character varying(50),
  exec_ref_id character varying(50),
  exec_inst character varying(50),
  exec_type character varying(4),
  exec_broker character varying(30),
  account character varying(30),
  clearing_firm character varying(30),
  clearing_account character varying(30),
  client_cross_ref character varying(30),
  symbol character varying(50),
  side character(1),
  price numeric(18,4),
  order_qty numeric,
  order_type character varying(4),
```

```

order_capacity character(1),
order_status character(1),
exec_transact_type character varying(4),
avg_price numeric(18,4),
last_price numeric(18,4),
last_fill numeric,
no_of_fills smallint,
last_capacity character(1),
cum_qty numeric,
leaves_qty numeric,
time_in_force character(1),
expire_time character varying(30),
min_qty numeric,
max_floor numeric,
trade_liq_ind character(1),
exec_restate_reason character varying(30),
undisclosed character(1),
shortsell_naked_qty numeric,
shortsell_covered_qty numeric,
shortsell_long_qty numeric,
no_self_trade character varying(50),
no_self_trade_order_no character varying(50),
remove_cross_from_clearing character(1),
id_source character(1),
security_id character varying(10),
security_exchange character varying(10),
price_improvement numeric(18,4),
mid_point_execution character(1),
exec_venue character varying(10),
no_trade_feat character(1),
origin_of_trans character varying(20),
intermediary_id character varying(10),
directed_wholesale_index character(1),
order_restriction character(1),
on_market_cross_type character varying(10),
on_market_trade_report_type character(1),
transact_time timestamp without time zone,
time_stamp timestamp without time zone,
CONSTRAINT chxa_order_pkey PRIMARY KEY (system_id, order_id)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE chxa_orders OWNER TO postgres;
GRANT ALL ON TABLE chxa_orders TO postgres;

```

Table – chxa_last_clord_id

This table is used to ensure a unique ClOrdId for each cancellation transaction. It is updated after each cancellation request to ensure each request has a unique id.

Fucntions:- chxa_get_cl_ord() and chxa_update_cl_ord()

```
-- Table: chxa_last_clord_id
-- DROP TABLE chxa_last_clord_id;

CREATE TABLE chxa_last_clord_id
(
  id bigserial NOT NULL,
  system_id uuid NOT NULL,
  last_clord_id integer,
  CONSTRAINT chxa_last_clord_id_pkey PRIMARY KEY (system_id)
)
WITH (
  OIDS=TRUE
);
ALTER TABLE chxa_last_clord_id OWNER TO postgres;
GRANT ALL ON TABLE chxa_last_clord_id TO postgres;
GRANT SELECT(system_id), UPDATE(system_id), INSERT(system_id),
REFERENCES(system_id) ON chxa_last_clord_id TO public;
```


Table – chxa_order_cancel_result

This table is updated with the results of each order cancel request. The intention is that this table will be a long term ‘audit trail’ of cancellation activity.

Function :- chxa_update_order_cancel() - updates this table and the chxa_trans table with the results of each order cancel request.

```
-- Table: chxa_order_cancel_result
-- DROP TABLE chxa_order_cancel_result;

CREATE TABLE chxa_order_cancel_result
(
    id bigserial NOT NULL,
    system_id uuid,
    user_name character varying(30),
    request_id character varying(40),
    clord_id character varying(50),
    account character varying(30),
    symbol character varying(50),
    side character(1),
    order_id character varying(50),
    org_clord_id character varying(50),
    order_status character(1),
    cancelled_count integer,
    error_text character varying(256),
    cancel_type character varying(10),
    cancel_time timestamp without time zone
)
WITH (
    OIDS=TRUE
);
ALTER TABLE chxa_order_cancel_result OWNER TO postgres;
GRANT ALL ON TABLE chxa_order_cancel_result TO postgres;
```

Table – chxa_trans

Web clients create an entry in this table each time they issue a cancellation request. MCOOrdersCHXA[COG] updates this table when the cancelation request result is known. It is intended that web clients will archive the contents of this table.

```
-- Table: chxa_trans

-- DROP TABLE chxa_trans;

CREATE TABLE chxa_trans
(
    id bigint NOT NULL DEFAULT nextval('trans_id_seq'::regclass),
    uid character(128),
    account character varying(30),
    order_id character varying(50),
    status character(1),
    ip character varying(256),
    "timestamp" time without time zone DEFAULT now(),
    sid character varying(128),
    date date DEFAULT now(),
    remote_user character varying(128)
)
WITH (
    OIDS=FALSE
);
ALTER TABLE chxa_trans OWNER TO postgres;
GRANT ALL ON TABLE chxa_trans TO postgres;

-- Index: "chxaOrderID"

-- DROP INDEX "chxaOrderID";

CREATE INDEX "chxaOrderID"
    ON chxa_trans
    USING btree
    (order_id);
```

9.2 Database Parameters:

DATABASE_NAME=Name of the database to access.

e.g DATABASE_NAME=webdb

DATABASE_SERVER=The machine which is the PostgreSQL database server.

e.g DATABASE_SERVER=rjlinuxlap

DATABASE_PORT=Port for the PostgreSQL database.

e.g DATABASE_PORT=5432

DATABASE_USER_ID=PostgreSQL database user.

e.g DATABASE_USER_ID=postgres

DATABASE_PASSWORD= PostgreSQL database user password*

e.g DATABASE_PASSWORD=rjexxxxxx

9.3 npgsql files:

The following files should reside in the same directory as MCOOrdersCHXA.exe:-

Mono.Security.dll

Npgsql.dll

These files are the “npgsql” .net data provider for PostgreSQL.

9.4 SQL Script Files:

The following files create database tables:-

- 1)CREATE TABLE system
- 2)CREATE TABLE system_state
- 3)CREATE TABLE chxa_last_clord_id
- 4)CREATE TABLE chxa_orders
- 5)CREATE TABLE chxa_trans
- 6)CREATE TABLE chxa_order_cancel_result
- 7)CREATE TABLE market_exchange
- 8)CREATE TABLE user_market_columns

The following files create database functions:-

- 1)FUNCTION get_system_info
- 2)FUNCTION update_system_state
- 3)FUNCTION chxa_get_cl_ord
- 4)FUNCTION chxa_update_cl_ord
- 5)FUNCTION chxa_update_order
- 6)FUNCTION chxa_update_order_cancel
- 7)FUNCTION insert_user_market_columns

Appendix 1

Order Feed:

Field Number	Field Name in Output	Presence	Fix Field Name	Fix Tag
1	Country	{ Always }	(Internal)	-
2	Exchange	{ Always }	(Internal)	-
3	Market	{ Always }	(Internal)	-
4	FirmID	{ Always }	SenderCompID	49 (Header)
5	TraderID	{ Always }	TargetCompID	56 (Header)
6	ClientID/ParticipantID	{ Not Always }	ClientID	109
7	MsgSeqNo	{ Always }	MessageSeqNo	34 (Header)
8	OrderID	{ Always }	OrderID	37
9	SecondaryOrderID	{ Iceberg Orders (in Trades) }	SecondaryOrderID	198
10	ClOrdID	{ Always }	ClOrdID	11
11	OrigClOrdID	{ Cancel/Replace }	OrigClOrdID	41
12	ExecID	{ Always }	ExecID	17
13	TransactID	{ Trades }	TransactID	6807
14	ExecRefID	{ Trade Cancel }	ExecRefID	19
15	ExecInst	{ Not Always }	ExecInst	18
16	ExecType	{ Always }	ExecType	150
17	ExecBroker	{ Not Always }	ExecBroker	76
18	Account	{ Not Always }	Account	1
19	ClearingFirm	{ Always }	ClearingFirm	439
20	ClearingAccount	{ Not Always }	ClearingAccount	440
21	ClientCrossRef	{ Not Always }	ClientCrossRef	6802
22	Symbol	{ Always }	Symbol	55
23	Side	{ Always }	Side	54
24	Price	{ Always }	Price	44
25	OrderQty	{ Always }	OrderQty	38

26	OrderType	{Not Always}	OrdType	40
27	OrderCapacity	{Not Always}	OrderCapacity	47
28	OrderStatus	{Always}	OrdStatus	39
29	ExecTransType	{Always}	ExecTransType	20
30	AvgPrice	{Always}	AvgPx	6
31	LastPrice	{Trade }	LastPx	31
32	LastShares	{Trade }	LastShares	32
33	LastCapacity	{Trade}	LastCapacity	29
34	CumQty	{Always}	CumQty	14
35	LeavesQty	{Always}	LeavesQty	151
36	TimeInForce	{Not Always}	TimeInForce	59
37	ExpireTime	{Not Always}	ExpireTime	126
38	MinQty	{Not Always}	MinQty	110
39	MaxFloor	{Not Always}	MaxFloor	111
40	PegDifference	{Not Always}	PegDifference	211
41	TradeLiqIndicator	{Trade}	TradeLiquidityIndicator	9882
42	ExecRestatementReason	{ Replace }	ExecRestatementReason	378
43	Undisclosed	{Not Always}	Undisclosed	6801
44	ShortSellNakedQty	{Not Always}	ShortSellNakedQty	6803
45	ShortSellCoveredQty	{Not Always}	ShortSellCoveredQty	6804
46	ShortSellLongQty	{Not Always}	ShortSellLongQty	6805
47	NoSelfTrade	{Not Always}	NoSelfTrade	8174
48	NoSelfTradeOrderNo	{Cancel}	NoSelfTradeOrderNum	8175
49	RemoveCrsFrmClearing	{Not Always}	RemoveCrossingFromClearing	8177
50	IDSsource	{Done for Day}	IDSsource	22
51	SecurityID	{Done for Day}	SecurityID	48
52	SecurityExchange	{Done for Day}	Security Exchange	207
53	PriceImprovement	{Trade Report}	PriceImprovement	639
54	MidPointExecution	{Trade Report}	MidPointExec	6833
55	ExecVenue	{Trade Report}	ExecVenue	4805
56	NoTradeFeat	{New/Cancel/Replace}	NoTradeFeat	7713
57	OriginOfTrans	{New/Cancel/Replace/Trade}	OriginOfTrans	8138

58	IntermediaryID	{New/Cancel/Replace/Trade}	IntermediaryID	8139
59	DirectedWholesaleIndex	{New/Cancel/Replace/Trade}	DirectedWholesaleIndic	8140
60	OrderRestriction	{New}	OrderRestrictions	8182
61	OnMarketCrossType	{On market trade execution report}	CrossType	8183
62	OnMarketTradeReportType	{On market trade execution report}	TradeReportType	8184
63	TransactTime (UTC)	{Always}	TransactTime	60
64	ExecutedDate (Local)	{Always}	Date(TransactTime)	-
65	ExecutedTime (Local)	{Always}	Time(TransactTime)	-
66	TimeStamp (UTC)	{Always}	TransactTime	-

Order Table:

Field Number	Column Name	Fix Field Name	Fix Tag
1	country	(Internal)	-
2	exchange	(Internal)	-
3	market	(Internal)	-
4	firm_id	SenderCompID	49
5	trader_id	TargetCompID	56
6	client_id	ClientID	109
7	message_no	MessageSeqNo	34
8	order_active	(Derived)	-
9	order_id	OrderID	37
10	secondary_order_id	SecondaryOrderID	198
11	clord_id	ClOrdID	11
12	org_clord_id	OrigClOrdID	41
13	exec_id	ExecID	17
14	transact_id	TransactID	6807
15	exec_ref_id	ExecRefID	19
16	exec_inst	ExecInst	18
17	exec_type	ExecType	150

MCOOrdersCHXA[COG]

Chi-X Australia Trading System Order Feed and Order Cancellation

18	exec_broker	ExecBroker	76
19	account	Account	1
20	clearing_firm	ClearingFirm	439
21	clearing_account	ClearingAccount	440
22	client_cross_ref	ClientCrossRef	6802
23	symbol	Symbol	55
24	side	Side	54
25	price	Price	44
26	order_qty	OrderQty	38
27	order_type	OrdType	40
28	order_capacity	Capacity	47
29	order_status	OrdStatus	39
30	exec_trans_type	ExecTransType	20
31	avg_price	AvgPx	6
32	last_price	LastPx	31
33	last_fill	LastShares	32
34	no_of_fills	(Derived)	-
35	last_capacity	LastCapacity	29
36	cum_qty	CumQty	14
37	leaves_qty	LeavesQty	151
38	time_in_force	TimeInForce	59
39	expire_time	ExpireTime	126
40	min_qty	MinQty	110
41	max_floor	MaxFloor	111
42	peg_diff	PegDifference	211
43	trade_liq_ind	TradeLiquidityIndicator	9882
44	exec_restate_reason	ExecRestatementReason	378
45	undisclosed	Undisclosed	6801
46	shorttsell_naked_qty	ShortSellNakedQty	6803
47	shortsell_covered_qty	ShortSellCoveredQty	6804
48	shortsell_long_qty	ShortSellLongQty	6805
49	no_self_trade	NoSelfTrade	8174

MCOrdersCHXA[COG]

Chi-X Australia Trading System Order Feed and Order Cancellation

50	no_self_trade_order_no	NoSelfTradeOrderNum	8175
51	remove_cross_from_clearing	RemoveCrossingFromClearing	8177
52	id_source	IDSource	22
53	security_id	SecurityID	48
54	security_exchange	SecurityExchange	207
55	price_improvement	PriceImprovement	639
56	mid_point_execution	MidPointExec	6833
57	exec_venue	ExecVenue	4805
58	no_trade_feat	NoTradeFeat	7713
59	origin_of_trans	OriginOfTrans	8138
60	intermediary_id	IntermediaryID	8139
61	directed_wholesale_index	DirectedWholesaleIndic	8140
62	order_restriction	OrderRestrictions	8182
63	on_market_cross_type	CrossType	8183
64	on_market_trade_report_type	TradeReportType	8184
65	transact_time_utc	TransactTime	60
66	time_stamp_utc	(Database Update Time)	-