HKEOrders

HKEx Orion Central Gateway ORDER Feed & SENTRY

This Document:

HKEOrders.pdf Rev 1.0

This document details how to install, configure and run HKEOrders

Revision:

04/11/2014 – Rev 1.0 - V.V. – Produced the first version of this manual.

1. Overview

HKEOrders communicates with HKEx's Orion Central Gateway (hereafter OCG) via two different FIX protocol-based messaging interfaces, namely Drop Copy Interface and Trading Interface. The program creates one or more source sessions to connect to the Drop Copy Interface and a single session to connect to the Trading Interface. A FIX (TCP/IP) connection is there underneath each source session.

The Drop Copy sessions extract orders from the Execution Reports and the Trading session is there to facilitate order cancellation. The program then provides the orders via a comma delimited order feed to the client applications and also store them in the database. HKEOrders allows SENTRY or other external clients to connect and routes their order cancel requests to the OCG.

The following diagram depicts the overall functionality and connectivity of the HKEOrders program in a production environment.

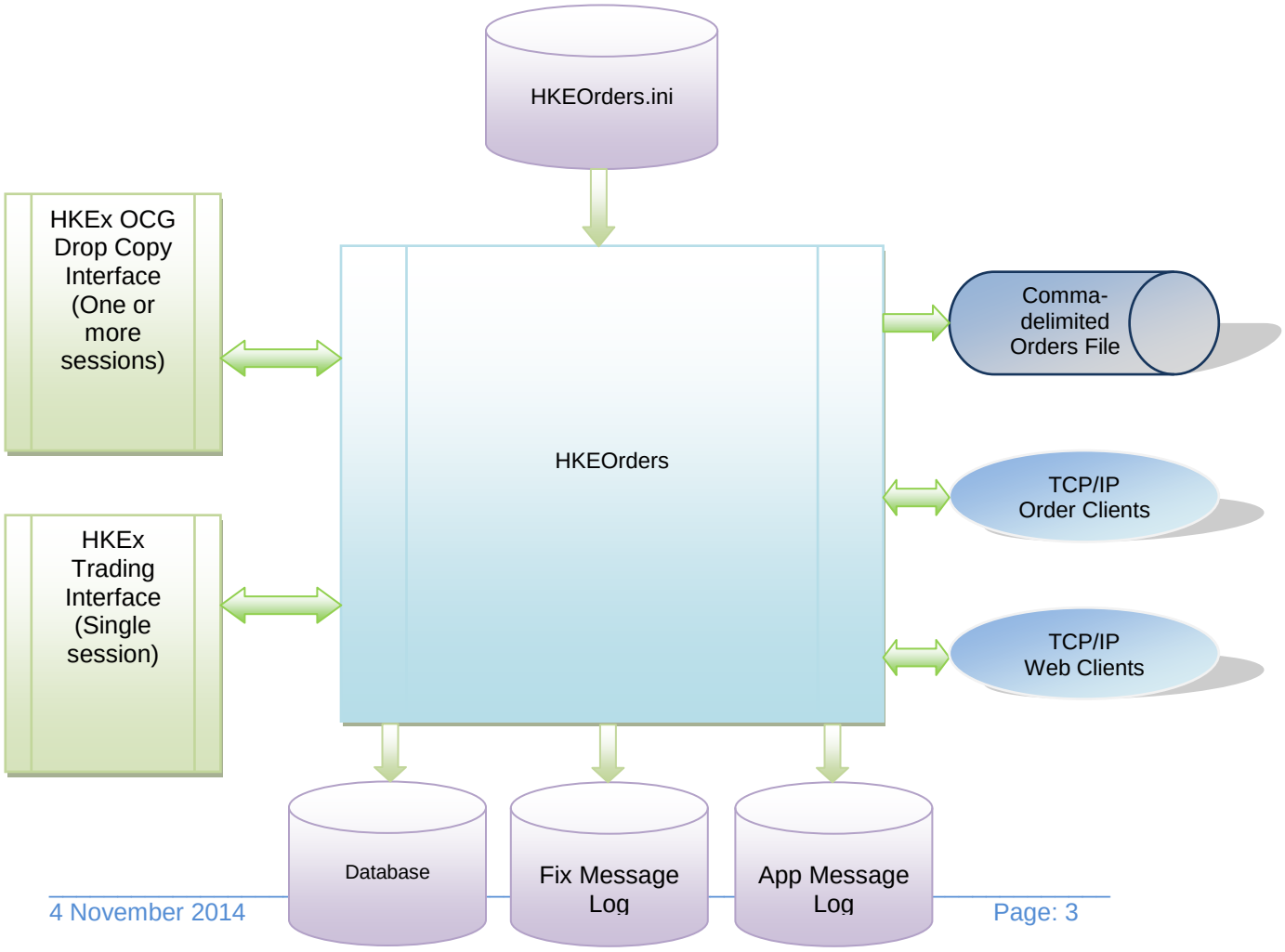


Figure 1. HKEOrders in a Production

1.1 Features:

Order Feed

Orders are available in the following output forms.

- Comma-delimited order file
- Comma-Delimited TCP/IP order feed

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

Data Store

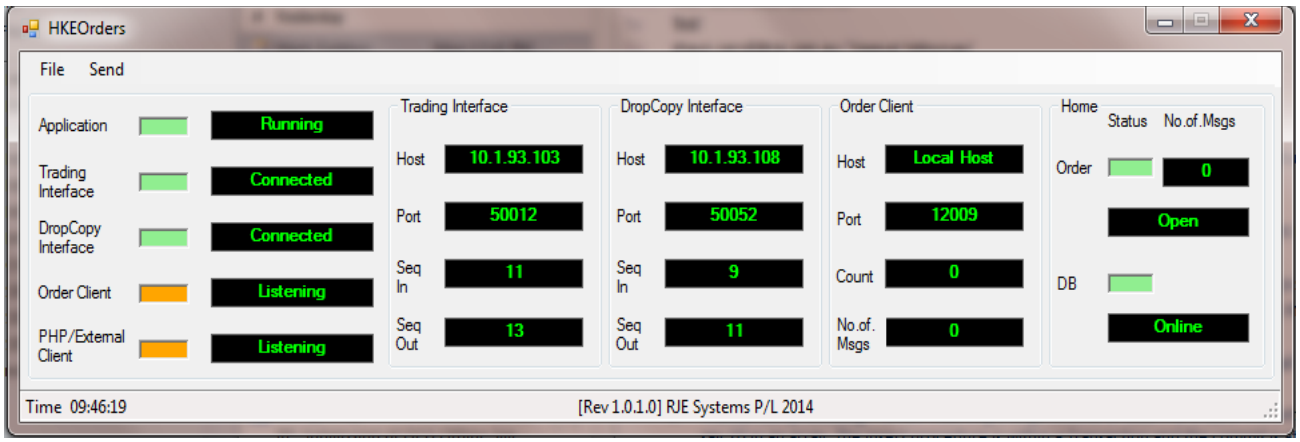
Orders extracted are stored in the database table order_record. A database is required for this feature. More details can be found in 8. Database

Order Cancellation

SENTRY (a PHP based web client) or another external client can connect and route their cancel requests to OCG via HKEOrders.

More details can be found in 7. Command Clients

1.2 GUI Screen:



The program's GUI 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.

Program Status:-

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

Trading Interface Status:-

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

Drop Copy Interface Status:-

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

Order Client Status: - (if accepting client connections)

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

PHP Client 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

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

Refer 4.7 Daily Cycle Parameters:

Note: We currently have no way of detecting Market Close through HKEx's OCG sessions. A timed shutdown is the only option.

3. Installation

3.1 Files and Directories:

Install HKEOrders as follows:-

<Install Directory>:- HKEOrders.exe, HKEOrders.ini
<Install Directory>:- Mono.Security.dll, Npgsql.dll
<Install Directory>/logs: - Make a subdirectory for log files
APP_LOG_DIRECTORY=logs.
FIX_LOG_DIRECTORY=logs.
<Install Directory>data :- make a subdirectory for data files.
APP_DATA_DIRECTORY=data

3.2 Password Encryption (.pem file):

The logon password the program sends to OCG must be encrypted with a public key supplied by OCG. A .pem file carries this information and it's specified in the following ini setting. The .pem file should be stored in the data directory.

PUBLIC_KEY_FILE=hkex_ocg_public_key.pem

3.3 Running HKEOrders:

In order to start running the program, run HKEOrders.exe, with the presence of a correctly configured HKEOrders.ini file, individual ini files for each session, and the two libraries Mono.Security.dll and Npgsql.dll.

You must set the following parameters correctly:-

- Parameters – OCG Session Configuration 4.1 FIX Session Parameters:
- Parameters – OCG Connection Configuration 4.2 FIX Connection Parameters:
- Parameters – OCG Logon Configuration 4.3 FIX 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

The main configuration parameters are stored in HKEOrders.ini

4.1 FIX Session Parameters:

The number of sessions is specified in the following parameter. This number includes all the Drop Copy sessions and a single Trading session.

NO_SESSIONS=(n)
e.g. NO_SESSIONS=1

For each session there is an additional ini file:- HKEOrders_Session(n).ini

e.g HKEOrders_Session1.ini, HKEOrders_Session2.ini

Note: One drop copy session is dedicated for gathering orders for one Exchange Participant also referred as EP. Configured with multiple drop copy sessions, one single HKEOrders program is able to gather data for multiple EPs.

Note: Only one order feed will be output regardless of the number of sessions configured.

4.2 FIX Connection Parameters:

Per session in HKEOrders_Session(n).ini

FIX_CONNECTION_POINTS = A set of IP address and port pair called Connection Points

e.g FIX_CONNECTION_POINTS=<host1>|<port1>,<host2>|<port2>,……

FIX_CONNECTION_POINTS=10.1.93.106|50050,10.1.93.107|50051,10.1.93.108|50052,10.1.93.109|50053,10.1.93.110|50054,10.1.93.116|50050,10.1.93.117|50051,10.1.93.118|50052,10.1.93.119|50053,10.1.93.120|50054

Note: The process involves Connection Points are called Hunt Connection Point process and HKEx supplies the list of the connection points to be used.

4.3 FIX Logon Parameters:

Per session in HKEOrders_Session(n).ini

FIX_SENDER_ID = Part of Fix header, a valid value must be specified.
e.g. **FIX_SENDER_ID**=CC15550002

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

FIX_PASSWORD = Password for Fix Logon.
e.g. **FIX_PASSWORD**=Dc140919

HKEx supplies the initial password which will be changed as needed by automatic password changing. Because of the rules for automatic password changing it is recommended the base is one upper case and one lower case letter.

FIX_PASSWORD_BASE = Base for automatic password changing (currently date is appended).
e.g **FIX_PASSWORD_BASE**=Dc

FIX_CHANGE_PASS = New Password
e.g. FIX_CHANGE_PASS=Aa123456

Note: Set this parameter to manually change the password. Please take note of HKEx's password rules.

4.4 Optional FIX Parameters:

FIX_HEARTBEAT=<Heartbeat interval> (Seconds) – default = 20
e.g. **FIX_HEARTBEAT**=10

Note: You should consult HKEx before setting this parameter, the default of 20 seconds is recommended.

4.5 Order Feed Parameters:

This is the TCP/IP port that the external clients can connect to receive an order feed derived from Execution Reports.

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

4.6 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**= HKEOrders
The name of the file e.g HKEOrders.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**= HKEOrders
The name of the file e.g HKEOrders.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.

4.7 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

4.8 Other Parameters:

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

See also

- 7.3 Command Clients Parameters:
- 8.2 Database Parameters:

4.9 Configuration File Example :

```
*******************************
* APPLICATION DETAILS         *
*******************************
APP_NAME=HKEOrders
INI_VERSION=1.0.1.0
APP_VERSION_POSTFIX=RJE Systems P/L 2014
*******************************
* WITH/WITHOUT GUI            *
*******************************
*NO_GUI=YES
*******************************
* FIX SESSION PARAMETERS      *
*******************************
NO_SESSIONS=2
*******************************
* CRYPTO                      *
*******************************
PUBLIC_KEY_FILE=hkex_ocg_public_key.pem
*******************************
* TCP CLIENTS' PARAMETERS     *
*******************************
ORDERS_PORT=12009
COMMAND_PORT=12010
*******************************
* MAXIMUM VALUES              *
*******************************
MAX_ORDER_CLIENTS=32
MAX_CMD_CONNECTIONS=120
MAX_CLORD_ID=9999999
*******************************
* APP LOG FILE PARAMETERS     *
*******************************
APP_LOG_FILE=HKEOrders
APP_LOG_DIRECTORY=logs
*******************************
* FIX LOG FILE PARAMETERS     *
*******************************
FIX_LOG_FILE=HKEOrders
FIX_LOG_DIRECTORY=logs
*******************************
* DATA FILE PARAMETERS        *
*******************************
APP_DATA_DIRECTORY=data
*******************************
* WAKE/SHUT TIMES             *
*******************************
WAKE_TIME=08:30
SHUT_TIME=20:00
*******************************
```

```
* DATABASE PARAMETERS         *
*******************************
DATABASE_NAME=hke
DATABASE_SERVER=127.0.0.1
DATABASE_PORT=5432
DATABASE_USER_ID=postgres
DATABASE_PASSWORD=rjeadmin
************ END **************
```

There is also a separate .ini file for each session. The above main configuration file indicates that there are two sessions, therefore there will be two sub .ini files.

```
*HKEOrders_Session1.ini
*======================
FIX_INTERFACE=DC
FIX_CONNECTION_POINTS=10.1.93.106|50050,10.1.93.107|50051,10.1.93.108|5
0052,10.1.93.109|50053,10.1.93.110|50054,10.1.93.116|50050,10.1.93.117|
50051,10.1.93.118|50052,10.1.93.119|50053,10.1.93.120|50054
FIX_SENDER_ID=CC15550001
FIX_TARGET_ID=HKEXCC
FIX_PASSWORD=Aa123456
FIX_PASSWORD_BASE=Dc
*FIX_CHANGE_PASS=Aa123456
*******************************
* SEQ NO                       *
*******************************
*SEQ_NO_OUT=0
*SEQ_NO_IN=0
************ END **************
```

```
*HKEOrders_Session2.ini
*======================
FIX_INTERFACE=TRADE
FIX_CONNECTION_POINTS=10.1.93.101|50010,10.1.93.102|50011,10.1.93.103|5
0012,10.1.93.104|50013,10.1.93.105|50014,10.1.93.111|50010,10.1.93.112|
50011,10.1.93.113|50012,10.1.93.114|50013,10.1.93.115|50014
FIX_SENDER_ID=CO15550003
FIX_TARGET_ID=HKEXCO
FIX_PASSWORD= Aa123456
FIX_PASSWORD_BASE=Dc
*FIX_CHANGE_PASS=Aa123456
*******************************
* SEQ NO                       *
*******************************
*SEQ_NO_OUT=0
*SEQ_NO_IN=0
************ END **************
```

5 Comma-Delimited Application Development

One option for developers is to make a TCP/IP connection to HKEOrders order feed port and receive order data in comma-delimited format. Data is simply sent when it is available; there is no need to request data.

The port for clients' connections is configured in 4.5 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:2,Exchange|S:4,Market|S:4,FirmID|S,TraderID|S,SeqNo|N,ClOrderID|S,OrigClOrdID|S,OrderID|S,ExecID|N:10,ExecRefID|N:10,ClientID|S,ExecBroker|S,ContraFirm|S,SecurityID|S,SecurityIDSource|N:10,SecurityExchange|S,OrdType|S,TimeInForce|S,Side|C,OrderQty|N,Price|N:12.7,TransactTime|TS,OrderCapacity|S,OrderRestrictions|S,MaxPriceLevels|S,PositionEffect|S,OrdStatus|N:1,ExecType|C,TradePrice|N:12.7,TradeQty|N:9,CumQty|N,LeavesQty|N:10,MatchType|S,TradeMatchID|S,OrderCategory|S,Text|S,LotType|S,ExecRestatementReason|S,CopyMsgIndicator|S,TradeDate|D,TradeTime|T,TimeStampUTC|TS,~

5.2 Comma-Delimited Data:

Fields that are not relevant are simply empty.

Orders

HK,HKEX,HKE,HKEXCC,CC15550001,767,1096,,17585,N.zb.S.OsedD8,,,2559,2558,11,8,XHKG,2,0,2,1000,100.2,20140910-06:17:52.000,,,,,2,F,100.2,1000,1000,0,4,11000000026,A,,2,,Y,20140910,16:17:52,20140910-06:17:52.000,~

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 order data. 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 HKEOrders.20140808.orders

6 Message Sequence Numbers

Each FIX session's Message Sequence Number starts 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.

6.1 FIX Message Log:

Each FIX session is continued across runs and there is a single FIX Message log for each session 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 4.6 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 Reset FIX Message Sequence Number:

HKEOrders should automatically recover from all issues relating to message sequence numbers.

It is strongly recommended that you use the automatic process where possible, by simply running the program.

In an extreme circumstance, it may be necessary to override the next expected Message Sequence Numbers; this can be done using program arguments as follows.

-so {SequenceNumberOut}
-si {SequenceNumberIn}

Note:

i) 'SequenceNumberOut' means the next Sequence Number of the message going out from HKEOrders to OCG
ii) 'SequenceNumberIn' means the next Sequence Number of the message coming in to HKEOrders from OCG

Note: This is to support a one-off run rather than adding parameters to the .ini file.

In an even more extreme circumstance, you can request that HKEx to reset the Message Sequence Numbers. But it is best to avoid it as some data may be lost if you do this.

7 Command Clients

7.1 Order Cancellation:

SENTRY (a PHP based web client) or another external client can connect and route their cancel requests to OCG via HKEOrders.

The following FIX messages are related to order cancellation

- Order Cancel Request (FIX Message F)
- Order Cancelled (FIX Message 8)
- Order Cancel Reject (FIX Message 9)

When an order cancel request succeeds an Order Cancelled 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.

HKEOrders stores the cancel results in 'order_cancel_result' table in the database.

7.2 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|~

7.3 Command Clients Parameters:

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

7.4 Cancellation Database Updates:

Table: - **last_clord_id**

This table is used to ensure a unique ClOrdId for each cancel request. The value is incremented by one after each cancel request.

Table: - **order_cancel_record**

SENTRY creates an entry in this table each time it issues a cancel request. HKEOrders updates the relevant entry in the table when the exchange sends back a response for the request. The contents of this table need to be archived for any future audits on order cancels.

Table: - **order_cancel_result**

The result of each order cancel request will be added to this table. This table will serve as a long term ‘audit trail’ of the order cancels done via SENTRY.

8 Database

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

The HKEOrders application uses the "npgsql" .net data provider for PostgreSQL. It calls ProgreSQL Functions (Stored Procedures) for database access and updating.

8.1 Database Tables:

Table - system

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

This table also 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 exchange.

Functions:-

1. get_system_info() create/retrieve system table information for a particular system.
2. update_system_state() - updates the state of the system

Table – order_record

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

Table – last_clord_id

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

Fucntions:-

1. get_cl_ord()
2. update_cl_ord()

Table – order_cancel_result

This table is updated with the results of each order cancel request. This table will serve as a long term 'audit trail' of the order cancels done via SENTRY.

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

Table – order_cancel_record

Web clients create an entry in this table each time they issue a cancellation request. HKEOrders updates this table when the cancelation request result is known.

8.2 Database Parameters:

DATABASE_NAME=Name of the database to access.
e.g DATABASE_NAME=hke

DATABASE_SERVER=The machine which is the PostgreSQL database server.
e.g DATABASE_SERVER=rjelinuxlap

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

8.3 npgsql files:

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

```
Mono.Security.dll
Npgsql.dll
```

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

8.4 SQL Script Files:

The following files create database tables:-

1)CREATE TABLE system
2)CREATE TABLE order_record
3)CREATE TABLE last_clord_id
4)CREATE TABLE order_cancel_result
5)CREATE TABLE order_cancel_record
6)CREATE TABLE user (used by SENTRY only)
7)CREATE TABLE order_display_fields (used by SENTRY only)
8)CREATE TABLE filter (used by SENTRY only)
The following files create database functions:-

1)CREATE FUNCTION get_system_info
2)CREATE FUNCTION update_system_state
3)CREATE FUNCTION get_cl_ord
4)CREATE FUNCTION update_cl_ord
5)CREATE FUNCTION hke_update_order
6)CREATE FUNCTION hke_update_order_cancel
7)CREATE FUNCTION get_order_fields (called by SENTRY only)
8)CREATE FUNCTION insert_order_display_fields (called by SENTRY only)

Note: Please refer SENTRY documentation for more details about the SENTRY only Postgre tables and function.

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	SeqNo	{Always}	MessageSeqNo	34 (Header)
7	ClOrdID	{Always}	ClOrdID	11
8	OrigClOrdID	{Cancel/Replace}	OrigClOrdID	41
9	OrderID	{Always}	OrderID	37
10	ExecID	{Always}	ExecID	17
11	ExecRefID	{Trade Cancel}	ExecRefID	19
12	ClientID	{Party Role = Location ID/BS User ID}	PartyID	448
13	ExecBroker	{Party Role = Executing Firm}	PartyID	448
14	ContraFirm	{Party Role = Contra Broker} for Trades	PartyID	448
15	SecurityID	{Always}	SecurityID	48
16	SecurityIDSource	{Always}	SecurityIDSource	22
17	SecurityExchange	{Always}	SecurityExchange	207
18	OrderType	{Not Always}	OrdType	40
19	TimeInForce	{Not Always}	TimeInForce	59
20	Side	{Always}	Side	54
21	OrderQty	{Not Always}	OrderQty	38
22	Price	{Not Always}	Price	44

23	TransactTime (UTC)	{Always}	TransactTime	60
24	OrderCapacity	{Not Always}	OrderCapacity	528
25	OrderRestrictions	{Not Always}	OrderRestrictions	529
26	MaxPriceLevels	{Not always}	MaxPriceLevels	1090
27	PositionEffect	{Not always}	PositionEffect	77
28	OrderStatus	{Always}	OrdStatus	39
29	ExecType	{Always}	ExecType	150
30	TradePrice	{Trade }	LastPx	31
31	TradeQty	{Trade }	LastShares	32
32	CumQty	{Always}	CumQty	14
33	LeavesQty	{Always}	LeavesQty	151
34	MatchType	{Not Always}	Match Type	574
35	TradeMatchID	{Trade}	TrdMatchID	880
36	OrderCategory	{Not Always}	OrderCategory	1115
37	Text	{Not Always}	Text	58
38	LotType	{Not Always}	LotType	1093
39	ExecRestatementReason	{Not Always}	ExecRestatementReason	378
40	CopyMsgIndicator	{Always}	CopyMsgIndicator	797
41	TradeDate (Local)	{Always}	Date(TransactTime)	-
42	TradeTime (Local)	{Always}	Time(TransactTime)	-
43	TimeStamp (UTC)	{Always}	TransactTime	-

Order Table:

Field Number	Colum 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	message_no	MessageSeqNo	34
7	clord_id	ClOrdID	11
8	org_clord_id	OrigClOrdID	41
9	order_id	OrderID	37
10	order_active	(Derived)	-
11	exec_id	ExecID	17
12	exec_ref_id	ExecRefID	19
13	client_id	PartyID (BSUserID)	448
14	exec_broker	PartyID (ExecBroker)	448
15	contra_firm	PartyID (ContraBrokerID)	448
16	security_id	SecurityID	48
17	security_id_source	SecurityIDSource	22
18	security_exchange	SecurityExchange	207
19	order_type	OrdType	40
20	time_in_force	TimeInForce	59
21	side	Side	54
22	order_qty	OrderQty	38
23	price	Price	44
24	transact_time	TransactTime	60
25	order_capacity	Capacity	528
26	order_restriction	OrderRestrictions	529
27	max_price_levels	MaxPriceLevels	1090

28	position_effect	PositionEffect	77
29	order_status	OrdStatus	39
30	exec_type	ExecType	150
31	last_price	LastPx	31
32	last_fill	LastQty	32
33	no_of_fills	(Derived)	-
34	cum_qty	CumQty	14
35	leaves_qty	LeavesQty	151
36	match_type	MatchType	574
37	trade_match_id	TrdMatchID	880
38	order_category	OrderCategory	1115
39	description	Text	58
40	lot_type	LotType	1093
41	exec_restate_reason	ExecRestatementReason	378
42	transact_time_utc	TransactTime	60
43	time_stamp_utc	(Database Update Time)	-