

Product Information:	2
This Document:.....	2
Revision:	2
1. Overview:.....	3
1.0 Program Operation:.....	3
1.1 OSEOrders:.....	4
1.2 Order Cancellation:	4
1.3 SQL Database:	4
1.4 Web Client:.....	4
2. Daily Cycle:.....	5
2.1 Trading Date:	5
3. Installation/Configuration:	6
3.1 Installation:	6
3.1.1 OSEOrders:.....	6
3.1.2 MCclickOSE:	6
3.2 OSEOrders Configuration:	7
3.2.1 MCclickOSE Connection Parameters:	7
3.2.2 J-Gate Logon Parameters:.....	7
3.2.3 Broker List Parameter:	8
3.2.5 Trades Data:.....	9
3.2.6 TCP/IP Feed Parameters:.....	9
3.2.7 Command Client Parameters – Order Cancellation.....	9
3.2.8 SQL Database Parameters	10
3.2.9 Logging Parameters:	10
3.2.10 Daily Cycle Parameters:.....	11
3.2.11 Other parameters:	11
3.2.12 New parameters:	11
3.3 MCclickOSE Configuration:	12
3.3.1 Client (OSEOrders) Connection Parameters:	12
3.3.2 J-Gate Connection Parameters:.....	12
3.3.3 Concurrent Broadcast Parameters:.....	12
3.3.4 J-Gate Connection Options:	12
3.3.5 Logging Options:	13
3.3.6 Other Configuration Options:	13
4. Password Changing:	14
5. Recovery Strategy:	14
6. Command Clients – Order Cancellation:	15
6.1 Order Cancellation Overview:	15
6.2 Supported Cancellation Types:.....	15
6.3 Canceling Groups of Orders:	17
6.4 Wiping All User Orders – MO448:	18
6.5 Wiping All Firm Orders – MO449:.....	19
6.6 Wiping All User Inactive Orders:.....	20
6.7 Wiping All Firm Inactive Orders:	21
7. SQL Database:	22
7.1 SQL Database Tables:.....	22
7.1.1 Table - system.....	22
7.1.2 Table – system_state	22
7.1.3 Table – ose_orders	22

7.1.4 Table – ose_order_cancel_result	23
7.2 SQL Database Parameters:	23
7.3 npgsql files:	23
7.4 SQL Script files:	24
8. PHP – Web Client:	24
9. Technical Information:	24
9.1 OM Click Queries/Broadcasts and Transactions:	24
9.1.1 Startup Queries	24
9.1.2 Transactions	24
9.1.3 Broadcasts	25
9.2 Concurrent Broadcasts:	25
10. Fields Mappings OSEOrders <-> J-Gate:	26

Product Information:

Please consult the RJE Web Site www.rje.com.au for the latest information on RJE products.

This Document:

OSEOrders – OSE J-Gate Orders Capture – details all issues relating to installation and operation of OSEOrders.

Revision:

12/04/2013 – Added JNet Orders, Trading Date – OSEOrders Rev 01_01.
25/03/2013 – Added Bulk Inactive order deletion – OSEOrders Rev 01_01.
06/03/2013 – Added Field Mappings– OSEOrders Rev 01_01.
29/11/2012 – Added MO448/MO449 – OSEOrders Rev 01_01.
22/11/2012 – Initial Document – OSEOrders Rev 01_01.

1. Overview:

1.0 Program Operation:

There is a separate MCclickOSE component, similar to OSEOrders.

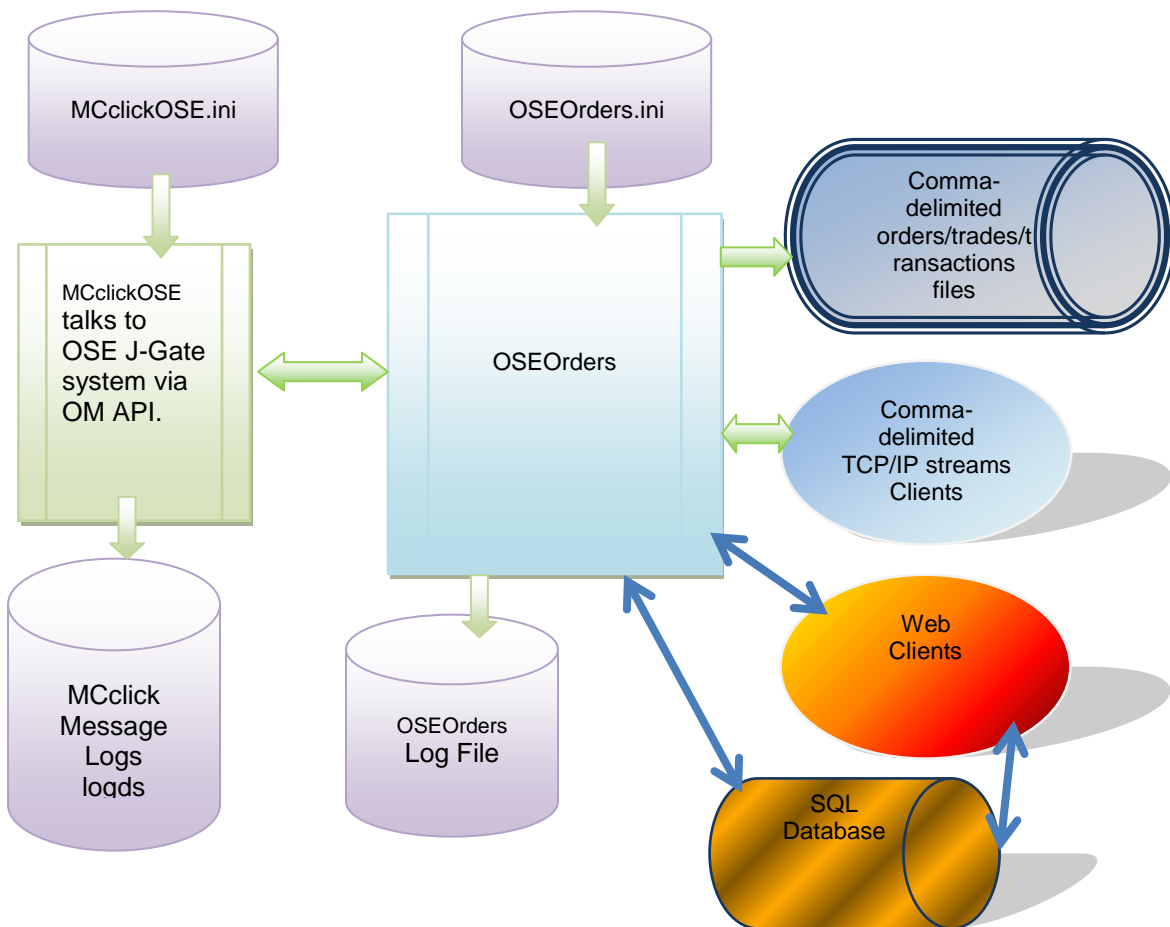


Figure 1. The OSEOrders Production System

1.1 OSEOrders:

The OSEOrders Program:-

- Connects to the OSE J-Gate system via MCclickOSE
- Updates the SQL database with details of current orders.
- Performs order cancellations in the J-Gate system via MCclickOSE.
- Provides TCP/IP stream feeds of orders, transaction and trades information.
- Provides file copies of orders, transaction and trades information.

The configuration settings for the system are found in OSEOrders.ini

1.2 Order Cancellation:

SFEOrders supports cancellation of orders via:-

- Web Clients
- GUI Clients

More details can be found in:- [6. Command Clients – Order Cancellation:](#)

1.3 SQL Database:

An optional SQL database can be configured to store orders and related information.

The SQL Database is required for Order Cancellation and for the Web Client.

More details can be found in :- [7. SQL Database:](#)

1.4 Web Client:

A PHP based web client application can be used for view / cancel orders.

See :- [8. PHP – Web Client:](#)

2. Daily Cycle:

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

Refer:- [3.2.6 Daily Cycle Parameters:](#)

WAKE_TIME = time when program wakes up each morning, SHUT_TIME = time when the program shutdown (hibernation) occurs. This area functions as per existing RJE products.

2.1 Trading Date:

Refer:- [3.2.6 Daily Cycle Parameters:](#)

The fields trading_date and t_plus_1 are determined by comparing the Order Creation Time to the T1_START time.

Existing orders which trade in the T1 session are also flagged as T1.

3. Installation/Configuration:

3.1 Installation:

3.1.1 OSEOrders:

Simply install OSEOrders as follows :-

<Install Directory> :- OSEOrders.exe, OSEOrders.ini

<Install Directory>:-au.com.mcx.dll, Npgsql.dll,

<Install Directory>/logs :- make a subdirectory for log files.

To run the program simply run OSEOrders.exe, provided the configuration in the .ini file is correct no other information is needed.

To connect to J-Gate MCclickOSE must also be running.

You must set the following parameters correctly:-

- 3.2.1 MCclickOSE Connection Parameters
- 3.2.2 J-Gate Logon Parameters
- 3.2.3 Broker List Parameter.

Note: If you wish to run the program without a GUI refer:-[3.2.11 Other parameters:](#)

3.1.2 MCclickOSE:

Typically, this would be in a separate directory but could be on the same machine.

Simply copy all files as follows;

<Install Directory> :- MCclickOSE.exe, MCclickOSE.ini

<Install Directory>:- libeay32.dll, oapimtdll.dll, ssleay32.dll, zlib1.dll

<Install Directory>/logs :- make a subdirectory for log files.

<Install Directory>:- plogOSE.exe – utility for converting log file extracts to text.*

* Messages to/from J-Gate are recorded in raw binary format. For some difficult to solve problems we may need to run this utility to view the actual data transmitted.

3.2 OSEOrders Configuration:

All configuration parameters are stored in OSEOrders.ini

3.2.1 MCclickOSE Connection Parameters:

MC_SERVER_HOST=myhost
MC_SERVER_PORT=7001

MC_SERVER_HOST = Name of Server running MCclickOSE.
e.g. MC_SERVER_HOST= myhost

MC_SERVER_PORT = Port to connect for MCclickOSE client connections.
e.g. MC_SERVER_PORT=7001

This corresponds to a port setting in MCclickOSE.ini:-
CLIENTS_PORT =7001

Note: OSEOrders always connects to the J-Gate system via MCclickOSE.

3.2.2 J-Gate Logon Parameters:

OSE_USER_ID= J-Gate Logon user supplied by the OSE.
e.g. OSE_USER_ID=2310000001

This should be a Trading for Admin user.

OSE_PASSWORD= J-Gate user password initially supplied by the OSE.
e.g. OSE_PASSWORD=NEXT55

OSE J-Gate user passwords can expire and must be changed periodically. OSEOrders will detect that a password has expired and will automatically change the password. The .ini file is updated with the new password; this is a full automated process and should not need manual intervention.

The following parameters can be set for automatic password handling:-

AUTO_PASSWORD_BASE=ABCDEF

CHANGE_PASSWORD_DAYS=10

CHANGE_PASSWORD_DAYS=n – change the password ‘n days’ before expiry.

3.2.3 Broker List Parameter:

You must define a list of broker numbers to be monitored by the OSEOrders application in this parameter. This setting ensures the OSEOrders program does MQ154 orders data queries for each broker in the list.

As well, the J-Gate system must be configured to correctly route orders and trades data for those brokers to the Trading for Admin user id.

This should always be tested, both querying existing orders via Broker List at start up and receiving details for any new orders via BO5 broadcasts.

BROKER_LIST=Brokers to query for MQ154.

* BROKER_LIST Data Processed for brokers in this list*

BROKER_LIST=23100,23101,23103,23104,23105,23106,23111,23113

3.2.5 Trades Data:

OSEOrders will capture trades information CQ10/BD6 and provide a separate trades feed.

ABN may also be getting a direct feed of trades information from the OSE Clearing System.

If necessary the trades feed from OSEOrders can be turned off.

```
*****  
* SET TRADES=NO if you don't want trades data *  
*****  
TRADES=NO
```

3.2.6 TCP/IP Feed Parameters:

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

ORDERS_PORT = TCP/IP port for all Orders.
e.g. **ORDERS_PORT=12008**

This is the primary feed produced by this application, but it also produces a feed of BO5 transactions and trades, this can be configured as follows:-

e.g. **ORD_TRANS_PORT=12012**
e.g. **TRADES_PORT=12014**

3.2.7 Command Client Parameters – Order Cancellation.

e.g. **COMMAND_PORT**= TCP port command clients must connect to issue cancel requests.
COMMAND_PORT=12010

3.2.8 SQL Database Parameters

The use of an SQL Database is optional but the Web Client facility will not work properly if it is not enabled.

All SQL Database access occurs in a separate thread it should not affect the performance of the rest of the OSEOrders application.

If we detect that database updating cannot keep up with the rate the OSE is sending data, the program can be enhanced with multiple database updating threads.

SQL_DATABASE_NAME=Name of the database to access. The presence of this parameter turns on database processing. All tables and functions mentioned in SQL_DATABASE_NAME=sfe

SQL_DATABASE_SERVER=The machine which is the PostgreSQL database server.
SQL_DATABASE_SERVER=rjlinuxlap

SQL_DATABASE_PORT=Port for the PostgreSQL database.
SQL_DATABASE_PORT=5432

SQL_USER_ID=PostgreSQL database user.
SQL_USER_ID=sfe

SQL_PASSWORD= PostgreSQL database user password*
SQL_PASSWORD=rjexxxxx

* There may be a better way of controlling access to the PostgreSQL database. We have chosen the user/password model to simplify the initial development.

3.2.9 Logging Parameters:

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= OSEOrders
filename= OSEOrders_20080429_150113.log.

APP_LOG_DIRECTORY=Directory where the application log is stored.
APP_LOG_DIRECTORY=logs

LOGGING_LEVEL= Set the level of application message logging; can turn on additional diagnostic messages.

LOGGING_LEVEL=9

3.2.10 Daily Cycle Parameters:

Refer:- [2. Daily Cycle:](#)

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

SHUT_TIME = time when the program shutdown (hibernation) occurs (hour:min)
default 23:30.
e.g. SHUT_TIME=21:00

T1_START_TIME/ T1_START_UTC = scheduled start of T+1 session, use to set trading date. T1_START_UTC is used if both are specified.

3.2.11 Other parameters:

NO_GUI=YES – Specify this value to run without a GUI, e.g. as a Windows NT service.

3.2.12 New parameters:

TRADES_UPDATE_ORDERS=NO – Setting this turns off updating orders with trades.

JNET_ORDERS=NO – Setting this removes JNet orders from the Orders feed.

In both cases above if nothing is specified the default is YES and the data is included.

T1_START_TIME/ T1_START_UTC = scheduled start of T+1 session, use to set trading date. T1_START_UTC is used if both are specified.

e.g.

T1_START_TIME=16:15

T1_START_UTC=07:15

3.3 MCclickOSE Configuration:

These are similar to parameter settings for similar components e.g. MCclickASX.ini, MCclickSGX.ini.

3.3.1 Client (OSEOrders) Connection Parameters:

As mentioned earlier OSEOrders must make a TCP/IP connection to MCclickOSE which in turn talks to the J-Gate system via OM API over a TCP/IP transport.

The corresponding settings for OSEOrders are:- [3.2.1 MCclickOSE Connection Parameters:](#)

CLIENTS_PORT = TCP/IP port that clients (in this case OSEOrders) connect to.
CLIENTS_PORT =7001

3.3.2 J-Gate Connection Parameters:

OSE_CLICK_GATEWAY = J-Gate Gateway to connect to – supplied by the OSE.
e.g. **OSE_CLICK_GATEWAY** =oseClick
*192.168.31.200 oseClick

OSE_CLICK_PORT =Port to use on J-Gate Gateway – supplied by the OSE.
OSE_CLICK_PORT =8026

3.3.3 Concurrent Broadcast Parameters:

For OSE J-Gate we must use Concurrent Broadcasts via the following .ini setting.

CONCURRENT_BROADCASTS=YES

If this line is omitted, we will use the older method of polling for broadcasts.

OSEOrders was tested with Concurrent Broadcasts.

3.3.4 J-Gate Connection Options:

These control the use of encryption or compression on the OM API link.

The OSE will advise if Compression or Encryption is to be used and our settings must match theirs.

e.g. **OMNI-API_COMPRESS** =NO (YES)

e.g. **OMNI-API_ENCRYPT** =NO (YES)

3.3.5 Logging Options:

These control the amount of information being logged.

DIAGNOSTIC_LEVEL =1 - controls how much information is logged in text diagnostics messages. Higher number mean more information is logged. Just use the default value unless otherwise instructed by RJE support personnel.

LOG_MESSAGES =A - controls amount of info logged

A=All, C=Client, X=Exchange, T=Text, W=Warning, E=Error, N=None

Can specify a single type or multiples (e.g. C+T)

A=C+X+T

T -> All Text messages includes warnings & errors.

W -> Warnings includes errors

Lowest setting is E -> Error messages only.

Error messages are always logged if logging is enabled.

N -> Turns logging off

Log files can get big quickly but logged info gives us the ability to diagnose problems.

LOG_FILTER - further control on amount of info logged for Exchange & MC API messages

D = log deals

5 = log BO5's

When the filter is set no other query response/broadcast message types are logged.

By default the filter is not set and all message types are logged.

Example – the recommended settings for MCTrades are:-

- **LOG_MESSAGES** =C+W
- **LOG_FILTER** =D5

3.3.6 Other Configuration Options:

TCPIP_CONNECTIONS =n Allow 'n' concurrent TCP/IP connect attempts (backlog),
Default = 5.

Note: BCAST_POLL_RATE and BCAST_HBEAT_POLL don't apply if using CONCURRENT_BROADCASTS.

BCAST_POLL_RATE=n - Broadcast Poll Rate - Polls per second (default = 10)
BCAST_HBEAT_POLL=n – Special poll rate for order entry apps not subscribed to any broadcasts (default = 1).

QUIT_DELAY=’n’ milliseconds - time to wait before closing client socket after sending quit response. Default = 200 M/S. (You should not need to use this.)

Performance Statistics:

BCAST_STATS=n - Output Broadcast Stats every 'n' seconds - zero default = no stats

4. Password Changing:

OSE J-Gate user passwords can expire and currently must be changed every 60 days. OSEOrders will detect that a password has expired and will automatically change the password to one that is compliant with the OSE rules. The .ini file is updated with the new password; this is a full automated process and should not need manual intervention.

See [3.2.2 J-Gate Logon Parameters](#): for more details.

5. Recovery Strategy:

Each order transaction has a unique key:- **OrderID+Symbol+Side**

This allows for OMX block order transactions, which are used in the J-Gate system.

OSEOrders performs the following recovery sequences

A. When the link to J-Gate Drops out:-

1. Close all client connections (kicking off, connected applications).
2. Reconnect to J-Gate gap fill transactions (MQ154) and trades (CQ10)
3. Clients should be able to reconnect and receive a feed as per normal.
4. Database upgrading to continue as per normal

* To be tested.

B. Program stopped/started BO5 transactions file retained:-

1. Wipe all Orders in the SQL Database
2. Load transactions and trades files.
3. Reconnect to J-Gate gap fill transactions (MQ154) and trades (CQ10)
4. Clients can reconnect and all feeds should be in the same sequence.
5. All Database updates are reapplied based on the order of the BO5 transactions file.*

* A secondary copy can be used while the primary is being rebuilt.

C. Program stopped/started BO5 transactions file is deleted, as per B. above except the sequence of the TCP/IP feeds will change. The same set of data will be sent and at the end of the process the set of orders in the database will be the same.

If OSEOrders detects any serious problem, it will just stop running.

6. Command Clients – Order Cancellation:

6.1 Order Cancellation Overview:

SFEOrders supports cancellation of orders via:-

- Web Clients – The final production system.
- GUI Clients – TestClient.exe – temporary fallback option (allows cancellation of groups of orders).

Configuration settings are found in:- [3.2.7 Command Client Parameters – Order Cancellation.](#)

6.2 Supported Cancellation Types:

Cancel requests and sent to the commands port us plain text strings.

```
CxlType->CxlActive==A|CxlInactive==I|CxlStop==S
```

Bulk Deletes:-

```
CxlUser==U // "MO448"  
CxlFirm==F // "MO449"  
CxlJnet==J // "MO74"  
CxlUserInactive==V // via multiple MO424 requests.  
CxlFirmInactive==W // via multiple MO424 requests.
```

```
CANCEL_REQUEST|CxlType={cancel type}|
    [Security=*ALL*|Side={bid/ask/*}
    [Country={country}]
    |Firm={firm id}|[Trader={trader id}]] <- whose/trading code.
    [Account={account}] <- whose
    [Customer={cust info}][Exchange={exchange info}]
```

Note: Security must be set to *ALL* for Bulk Deletes. You cannot do this via the a separate GUI program TestClient.exe can be used to perform these deletes.

Delete a single order:-

```
CANCEL_REQUEST|CxlType={cancel type}|[OrderID={order id}
    [Security={security}]|Side={bid/ask}|
    [Country={country}]
    |Firm={firm id}|[Trader={traderid}] <- trading code.
```

Delete a group of orders:-

```
CANCEL_REQUEST|CxlType={cancel type}|
    [Security={security}]/Series*(1)|Side={bid/ask/*}
    [Country={country}]
    |Firm={firm id}|[Trader={trader id}]] <- whose/trading code.
    [Account={account}] <- whose
    [Customer={cust info}][Exchange={exchange info}]
```

Note: The absence of an order_id indicates a group cancel request.

There is also a simplified type of cancellation intended for use by the web client.

```
CANCEL_REQUEST| OrderID={order id}
```

In that mode OSEOrders will look up the order in memory and fill in any other fields required by the J-Gate system.

OSEOrders also supports the following more technically correct method of cancelling orders:-

```
CANCEL_REQUEST| UniqueKey={uniqueKey}
```

The UnqueKey field is :- OrderNumber+SeriesName+Bid/Ask

This is the correct way of cancelling orders that have been entered via block transactions or as quotes.

The Web Client should be amended to support this method of cancellation, use the unique_key field in the ose_orders table.

6.3 Canceling Groups of Orders:

It is possible to cancel groups of orders by following the rules of the MO4 transaction as documented in the relevant J-Gate manual.

As example of how to do this type of cancellation can be seen on the screen below.

The screenshot shows a web client window titled "Cancel Request". The form contains the following fields and values:

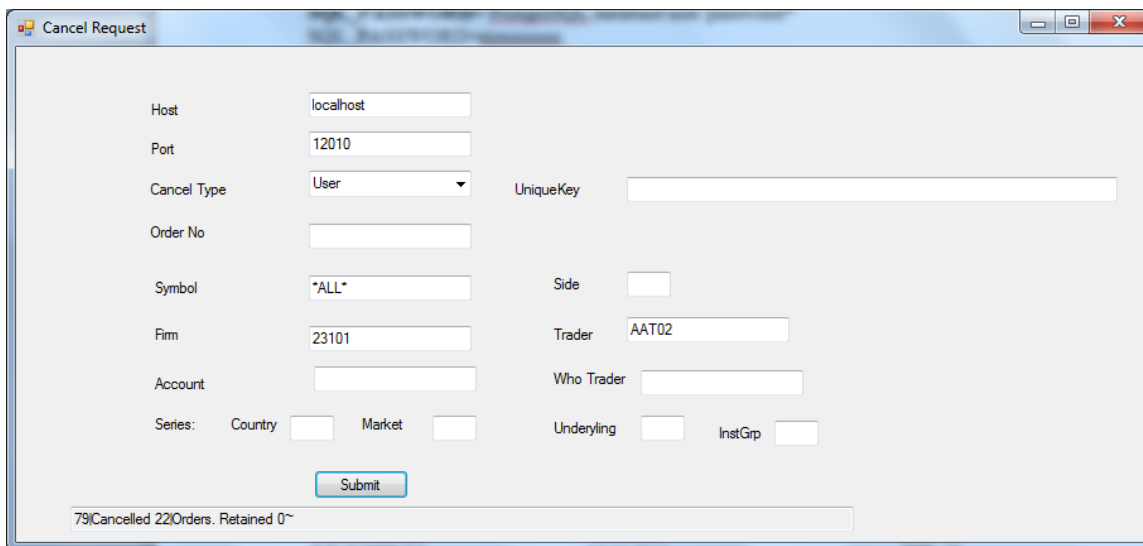
- Host: localhost
- Port: 12010
- Cancel Type: Active (dropdown menu)
- Order No: (empty)
- Symbol: CAL_1812_1212_200
- Firm: 23101
- Account: (empty)
- Series: Country [], Market []
- UniqueKey: (empty)
- Side: S
- Trader: AAT03
- Who Trader: (empty)
- Underlying: []
- InstGrip: []

A "Submit" button is located at the bottom center of the form. At the bottom of the window, a status bar displays the text "61|Cancelled 3|Orders.~".

```
2012-11-22 12:35:32 |CmdClient:CANCEL_REQUEST|USER=MARK|REQUEST_NO=ebbb01f6-6fae-4565-87c1-1e05d818ce74|CxlType=A|OrderID=|Symbol=CAL_1812_1212_200|Side=S|Country=OX|Firm=23101|Trader=AAT03|Account=|~
2012-11-22 12:35:32 |CmdClient:SEND:61|Cancelled 3|Orders.~
```

6.4 Wiping All User Orders – MO448:

This generate a number of sub commands to delete orders in each partition, the total number of orders deleted can be determined by adding the counts in the relevant rows in the database table “ose_order_cancel_result”.



Cancel Request

Host: localhost

Port: 12010

Cancel Type: User

UniqueKey: [Empty]

Order No: [Empty]

Symbol: *ALL*

Side: [Empty]

Firm: 23101

Trader: AAT02

Account: [Empty]

Who Trader: [Empty]

Series: Country [Empty] Market [Empty]

Underlying: [Empty] InstGrp: [Empty]

Submit

79/Cancelled 22/Orders. Retained 0~

```

2012-11-29 12:16:14 |CmdClient:CANCEL_REQUEST|USER=MARK|REQUEST_NO=178d86a3-94dc-4f4c-9eb7-
a1edaa5afeaa|CxlType=U|OrderID=|Symbol=*ALL*|Side=|Country=OX|Firm=23101|Trader=AAT02|Account=|~
2012-11-29 12:16:14 |SendClientRequest - RequestID: 79
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=P|SNA=SPF_DJIA|OBC=1|WHO=OX-23101-AAT02-|
TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:16:14 |SendClientRequest - RequestID: 80
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=P|SNA=SPO_5020|OBC=1|WHO=OX-23101-AAT02-|
TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:16:14 |SendClientRequest - RequestID: 81
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=P|SNA=SPF_RNP|OBC=1|WHO=OX-23101-AAT02-|
TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:16:14 |SendClientRequest - RequestID: 82
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=P|SNA=SPF_NK225M|OBC=1|WHO=OX-23101-AAT
02-|TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:16:14 |SendClientRequest - RequestID: 83
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=P|SNA=SPF_NK225|OBC=1|WHO=OX-23101-AATO
2-|TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:16:14 |SendClientRequest - RequestID: 84
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=P|SNA=SPO_NK225|OBC=1|WHO=OX-23101-AATO
2-|TCD=OX-23101-AAT02|BOS=|

```

Note: Be careful using this function, a confirmation screen now pops up as an additional warning that all active orders will be deleted.

6.5 Wiping All Firm Orders – MO449:

This generate a number of sub commands to delete orders in each partition, the total number of orders deleted can be determined by adding the counts in the relevant rows in the database table “ose_order_cancel_result”.

```

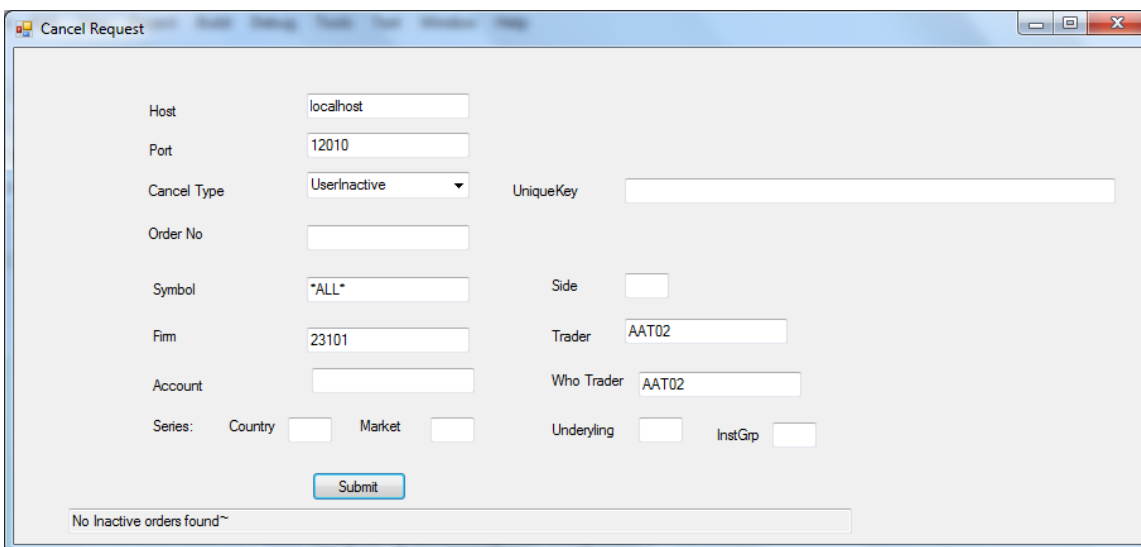
2012-11-29 12:32:11 |CmdClient:CANCEL_REQUEST|USER=MARK|REQUEST_NO=603bc358-1df0-42ff-b29e-
beb5e59dc7fe|CxIType=F|OrderID=|Symbol=*ALL*|Side=|Country=OX|Firm=23101|Trader=AAT02|Account=~
2012-11-29 12:32:11 |SendClientRequest - RequestID: 85
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=F|SNA=SPF_DJIA|OBC=1|WHO=OX-23101-AAT02-
TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:32:11 |SendClientRequest - RequestID: 86
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=F|SNA=SPO_5020|OBC=1|WHO=OX-23101-AAT02-
TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:32:11 |SendClientRequest - RequestID: 87
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=F|SNA=SPF_RNP|OBC=1|WHO=OX-23101-AAT02-
TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:32:11 |SendClientRequest - RequestID: 88
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=F|SNA=SPF_NK225M|OBC=1|WHO=OX-23101-AAT
02-|TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:32:11 |SendClientRequest - RequestID: 89
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=F|SNA=SPF_NK225|OBC=1|WHO=OX-23101-AATO
2-|TCD=OX-23101-AAT02|BOS=|
2012-11-29 12:32:11 |SendClientRequest - RequestID: 90
|MCT_FUNC=SEND|MSG=MSG_C_DELETE_ORDER|CODE=1640|DCY=F|SNA=SPO_NK225|OBC=1|WHO=OX-23101-AATO
2-|TCD=OX-23101-AAT02|BOS=|

```

Note: Be careful using this function, a confirmation screen now pops up as an additional warning that all active orders will be deleted.

6.6 Wiping All User Inactive Orders:

This generates a number of sub commands to delete orders in each series. OSEOrders must do a search though all orders in memory to determine which series have inactive orders. The total number of orders deleted can be determined by adding the counts in the relevant rows in the database table “ose_order_cancel_result”.



```
2013-03-22 16:55:12 |CmdClient:CANCEL_REQUEST|USER=MARK|REQUEST_NO=c36f0fad-ab93-4c7a-bc25-
dfd07a150508|CxIType=v|OrderID=|Symbol=*ALL*|Side=|Country=OX|Firm=23101|Trader=AAT02|Account=|WhoTrader
=AAT02|~
2013-03-22 16:55:20 |CmdClient:SEND:No Inactive orders found~
```



6.7 Wiping All Firm Inactive Orders:

This generates a number of sub commands to delete orders in each series. OSEOrders must do a search though all orders in memory to determine which series have inactive orders. The total number of orders deleted can be determined by adding the counts in the relevant rows in the database table “ose_order_cancel_result”.

Cancel Request

Host: localhost

Port: 12010

Cancel Type: FirmInactive

Order No:

Symbol: *ALL*

Firm: 23101

Account:

Series: Country Market

UniqueKey:

Side:

Trader: AAT02

Who Trader:

Underlying: InstGrp:

Submit

Error!Click Session Not Ready~

```
2013-03-25 10:10:03
|CmdClient:CANCEL_REQUEST|USER=MARK|REQUEST_NO=3be111b3-52f1-441d-81b6-
73f7e21348ae|CxlType=W|OrderID=|Symbol=*ALL*|Side=|Country=OX|Firm=23101|Trader=AAT
02|Account=~
```

7. SQL Database:

The database is common to a number of order pulling applications which gather orders information for use by the web client application.

The OSEOrders program uses the “npgsql” .net data provider for PostgreSQL.

The OSEOrders program typically calls PostgreSQL Functions (Stored Procedures) for database access and updating. This approach was chosen as we believe it will deliver the best performance.

7.1 SQL Database Tables:

7.1.1 Table - system

Standard Web Client table – used by OSEOrders.

7.1.2 Table – system state

Standard Web Client table – used by OSEOrders.

7.1.3 Table – ose orders

This is the main table of interest to Web Clients and other application which wish to display data.

When the field order_active='Y' the order is an active order which is a candidate for cancellation.

When the field order_active='I' the order is an inactive order which is a candidate for cancellation.

As orders trade out or are cancelled order_active is set to 'N'.

On a restart/rollover orders are deleted via `func_ose_wipe_current_orders()`

Function :- `func_ose_update_order()` – Updates the orders table for each execution report transaction. Note: It is likely that this function will be specific to the SFE other systems will use the same database table but may have a slightly different update function.

Function :- `func_ose_wipe_current_orders()` – Deletes all orders if required.

```
CREATE TABLE ose_orders
(
  id bigserial NOT NULL,
  system_id uuid NOT NULL,
  order_id character varying(50) NOT NULL,
  message_no integer,
  order_active character(1),
  order_status character varying(4),
  order_type character varying(4),
  order_book_class character varying(4),
  ord_change_reason character varying(4),
  order_ref character varying(50),
  order_modified_time timestamp without time zone,
  firm_id character varying(31),
  trader_id character varying(32)
,....
```

7.1.4 Table – ose 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 :- `func_ose_update_order_cancel()` - updates this table and the trans table with the results of each order cancel request.

7.2 SQL Database Parameters:

See [3.2.8 SQL Database Parameters](#)

7.3 npgsql files:

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

07/07/2007	12:09 AM	282,624	Mono.Security.dll
28/09/2011	08:55 PM	365,568	Npgsql.dll

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

7.4 SQL Script files:

The following files create database tables:-

19/11/2012	10:18 AM	2,177	ose_create_orders.sql
28/06/2012	10:12 AM	999	ose_create_order_cancel_result.sql

The following files create database functions:-

20/11/2012	11:24 AM	9,522	func_ose_update_order.sql
28/06/2012	10:13 AM	4,425	func_ose_update_order_cancel.sql
28/06/2012	10:02 AM	742	func_ose_wipe_current_orders.sql

8. PHP – Web Client:

This is documented elsewhere.

9. Technical Information:

9.1 OM Click Queries/Broadcasts and Transactions:

9.1.1 Startup Queries

Delta Queries:- DQ120/DQ122/DQ124/DQ126

Other DQ Queries :- DQ3, DQ7, DQ8, DQ29

Other startup:- UI1, UQ15

Trades :- CQ10

BO5 Transactions :- MQ154

9.1.2 Transactions

Transactions:- (“Trading (for Admin)”)

Order Deletes:- MO388,MO448,MO449,MO424, MO428, M040

For MO448/MO449 see [6.2 Supported Cancellation Types:](#)

9.1.3 Broadcasts

Broadcasts:-

BI41, BO5, BD6, BU124 (Series updates)

9.2 Concurrent Broadcasts:

Concurrent Broadcasts – the ability to receive Concurrent Broadcasts is a J-GATE requirement. This can be turned on/off via MCclickOSE.ini.

10. Fields Mappings OSEOrders <-> J-Gate:

OSE Orders Field Name	J-Gate Field Name
firm_id	trading_code.ex_customer_s
crader_id	trading_code.user_id_s
crder_id	order_number_u
cl_order_id	From exchange_info_s
exec_id	(last trade.trade_number)
exec_trans_type	<No Value>
order_status	order_state_u
order_bos_pos	ob_position_u
account	ex_client_s
exchange_code	<Always "OSE">
symbol	Series Name – (series.ins_id_s)
Side	bid_or_ask_c
order_qty	mp_quantity_i
Price	premium_i
last_shares	(last trade.quantity)
cum_quantity	<No Value>
transact_time	<No Value>
Text	customer_info_s
order_type	order_type_c
expire_time	time_validity_n.duration
commodity	<No Value>
Month	<No Value>
Year	<No Value>
ob_class	<RJE Derived field>
block	block_n
change_reason	change_reason_c
combo_mark	combo_mark_c
display_qty	display_quantity_i
exch_ord_type	exch_order_type_n
ex_state	ext_t_state_c
give_up_member	give_up_member
limit_premium	Stop orders – limit_premium_i
max_rand_hidden	mp_max_random_hidden_i
oc_request	open_close_request_c
order_no_bin	order_number_u

Party	Party
sequence_no	sequence_number_u
Series	Series
stop_condition	stop_condition_c
stop_series	Stop Orders – stop_series
stop_series_name	Stop Orders -(stop_series.ins_id_s)
time_val_type	time_validity_n.type
total_volume	total_volume_i
trading_code	trading_code
transaction_no	Transaction_number_n
user_code	ex_user_code
trans_ack	(Txstat from last BO5 transaction)
Active	<RJE Derived field>
acc_type	From exchange_info_s
tr_purpose	From exchange_info_s
instance	instance_c
unique_key	<RJE Derived field>
Created	timestamp_in
Modified	execution_timestamp
created_utc	timestamp_in
modified_utc	execution_timestamp
txn_type	<RJE Derived field>
omx_txn_code	e.g. MA154, BO5
omx_click_code	e.g. MO31 (transaction_no)
t_plus_1	<RJE Derived field> *1
jnet_order	<RJE Derived field>
trading_date	<RJE Derived field> *1

*1 See [2.1 Trading Date](#):