Scrutiny Order Monitoring Software

Technical Overview

This Document:

RJE Scrutiny Technical.pdf – Overview of Technical information.

Revision:

05/03/2015 - M.G. – Produced the first version of this manual
10/03/2015 – V.V. – Produced the client overview
13/04/2015 – M.G. – Revised
21/10/2015 – B.C. – Revised
02/08/2021 – C Carroll – Manual Update

1. Scrutiny Overview:

RJE Scrutiny is an RJE Systems application that monitors, checks, captures and notifies any irregularities in a firm's order transactions. Initially RJE developed **Scrutiny** for the ASX ASXTrade platform, however **Scrutiny** can be interfaced to any of the exchanges where RJE provides an order capture software product.

Scrutiny – ASXTrade users require an OMAPI connection to the ASXTrade system to extract data from the system.

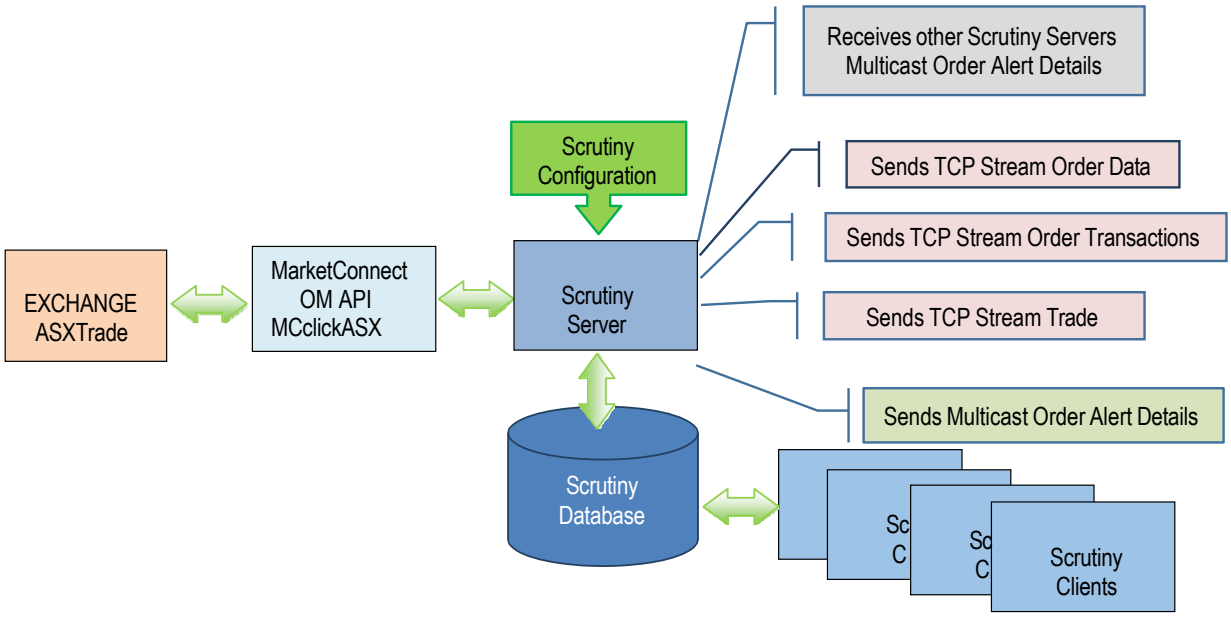


Figure 1. **Scrutiny** Overview

2. Features Overview:

2.1 Rules Checking for Order Transactions:

For the ASX Trading System ASXTrade, **Scrutiny** captures order transactions and trade records if required.

Predefined Rules are checked for all order transactions, even when **Scrutiny** may have been offline at the time the order transaction was entered.

Order transactions are sorted into time sequence so that as far as possible a version of **Scrutiny** recovering for an outage will get identical results to a version that was online while all transactions were entered.

2.2 Extendable Rule Checking:

One design aim was to allow checking to be extended as needed at particular customer sites. To achieve that aim we decided to define the rules in SQL.

The question then was if SQL based rule checking could deliver adequate performance. RJE has done extensive benchmarking on performance and is satisfied **Scrutiny** delivers adequate performance.

In summary the data required for SQL based rules checking, is stored in the database in an optimized, convenient form that allows the delivery of adequate performance.

As a result of storing this data the system is:-

- **Open**
- **Transparent**
- **Auditable**

It is easy to see what rules are being checked and what data is used to check those rules and it is possible to independently verify or analyze the data.

2.3 Adding Additional Rules – Parallel Copies:

New rules can be added via editing SQL stored procedures (functions).

Obviously care must be taken when doing this, as any SQL that queries a large number of rows is likely to be slow and slow down rule checking. However, fine tuning of the rules using data already present in the various tables delivers high performance.

The best way to add additional rules is to initially run a secondary copy of **Scrutiny** that can be feed via a TCP/IP stream of order transactions from the primary **Scrutiny** copy.

There is no need for this secondary copy to have a connection to the ASX, it should have no significant performance impact on the primary copy. This secondary copy will have its own database and can check a different set of rules.

Alerts from multiple **Scrutiny** applications can be consolidated via TCP/IP multicast messages. This would allow all alerts to be visible in all environments regardless of the source application/environment (at this stage we generate the TCP/IP multicast messages, but do not action them).

3. Database Tables:

Scrutiny captures and stores order and trade details in a PostgreSQL database. Predefined user limits are also set and stored in the database tables and are used for checking against orders transactions.

4. TCP/IP Feed Details:

4.1 Order Transactions:

ORD_TRANS_PORT=12012

4.2 Order Information (Rolling update):

ORDERS_PORT=12008

4.3 Trades:

TRADES_PORT=12014

4.4 Multicast Alerts:

MCAST_GROUP_ADDRESS=239.195.0
.32 **MCAST_PORT_NO**=16555
MCAST_HOPS=1
MCAST_ALERTS_ONLY=YES

5. Rules Checking:

For both Order Limits and Transaction limits we pre-accumulate the data required to check the rules into the database total records.

5.1 Order Limits:

Function: - check_order_limits ()

Order limits can be defined at any point in the following hierarchy:-

- **System**

- **System/Firm**
- **System/Firm/Trader**
- **System/Firm/Trader/Symbol**

These limits are defined in the user limits table and can be fix limits or another application could compute the daily limits for each trader and update the table.

Limits can be of various types (all examples below for order value):-

- **Order Value** = This order value
- **Trade Value** = This Trade value
- **Overall Value** = (total buy and sell orders + total buy and sell trades)
- **Net Value** = (total buy orders and trades – total sell orders and trades)
- **In Market Value** = (total buy orders value + total sell orders

value) Limits can be checked at the following arbitrary levels:-

- **BTCS** = Broker / Trader / Client / Stock
- **BTC** = Broker / Trader / Client
- **BTS** = Broker / Trader / Stock
- **BT** = Broker / Trader
- **BC** = Broker / Client
- **BS** = Broker / Stock

Note: In theory we can define and check limits at any level of the hierarchy, but the most common level would be BT – Broker Trader.

5.1.1 Checking limits against overall totals

This alert is for a single order value, but our rule example checks against all orders / trades for the broker.

Alert: Large Order vs Capital
Scrutiny will track market value of order vs a defined value which is determined relative to their capital/credit line and raise an alert when order market value exceeds that defined value $XXXX.

5.1.2 Order Value

This is a simple check on an order against a fixed limit

Alert: Large Order vs Set Value
Scrutiny will track order amount (shares or lots) vs a set limit and raise an alert when the order volume XXXX exceeds that defined $value XXXX.

5.2 Transaction Limits:

Function: - check_trans_limits ()

5.2.1 Transaction Rate Overall

Scrutiny accumulates into the order_totals record in the following fields:-
trans_per_sec

Total Transactions in the last 5,10,15,20,25,30,35,40,45,50,55,60 seconds

Alert: Rapid Order Entry
Scrutiny will raise an alert when Client suddenly inserts XXXX orders per XXXX seconds.

5.21 Transaction Rate Duplicate Orders

Totals are the accumulated into the order_totals record in the following fields:-
trans_stk_vol_price, Key – User+Stock – Last 1,000 transactions (type, timestamp, price, and volume), overall transaction totals and totals for each type (enter, amend, cancel, etc).

Alert: Duplicate orders
Scrutiny will raise an alert when the same order entry, amendment deletion i.e. by volume/price are being sent in a frequency of XXXX in YYYY seconds.

Equities: Using (Symbol + Volume + Price) total the number of orders by client over the XXXX second period

Note: The main overhead on this process is gathering the statistics.

5.3 Secondary Server:

You can configure a secondary server which can have a separate database on another machine.

This can connect to the primary Scrutiny Server and retrieve a feed of order transactions that it can use for rules checking.

It is strongly recommend that you use a secondary server for testing major rule changes, particularly new user manual rules.

Secondary settings:-

```
******************************
* Remote Scrutiny Server     *
******************************
```

REMOTE_HOST=ScrHost
REMOTE_PORT=12012

Primary settings:-

ORD_TRANS_PORT=12012

Note: Running 1 or more secondary servers should have little performance impact on the primary.

5.4 Defining Manual Rules:

Users can create manually defined function and they take the following form:-

CREATE OR REPLACE FUNCTION rje_manual_example_1
(uniquekey character varying, p1 integer, p2 numeric)

5.5 Guidelines for Manual Rules Development:

RJE has benchmarked the performance of automatically generate rules classes:

1. ValueLimit – rules automatically generated – limits in User Limits:
2. TransCount– rules automatically generated – parameters in rules table.

All of these rules check a single row in the database table and we have gone to considerable effort to ensure all of the required totals are pre-accumulated.

We want some flexibility in defining rules without the need for program changes, but we don't want to impact performance.

However, slower rule checking can always be done in a secondary server.

6. Archiving:

6.1 Automatic Archiving:

Scrutiny captures and stores the order, order transaction, trade, alerts and transaction totals on a daily basis. Therefore it is necessary to ensure at the start of each day the transactional tables are empty. This is achieved by copying the previous day's data to history tables.

When starting up each morning or on a program restart Scrutiny checks for the need to archive data.

If un-archived data is found then it is archived before any new order transactions for the day are processed. This is intended as a "catch-all" if scheduled archiving is not set or of the program is not running for some reason at the scheduled archiving time.

6.2 Scheduled Archiving:

- The recommended process is to schedule archiving for a particular time each day by setting the ARCHIVE_TIME parameter in ScrutinySvrASX.ini.
- **ARCHIVE_TIME**=18:00
- **SHUT_TIME**=19:00

Obviously, we need complete archiving before shutting down for the day, but any partially completed archiving should be completed automatically the next day.

When scheduled archiving is being done archiving of data in the history tables for orders that have existed for a week or more is also done.

Note: We still wipe data in some instances even though we are archiving, this is necessary for recovery synchronization purposes.

7. Scrutiny Client

Scrutiny has a user interface that allows users to easily interact with the **Scrutiny** module data.

The main features available in the client interface are:

- **Configuration of limits and rules**
- **Maintenance of limits and rules**
- **Notification/display of alerts**
- **Display of order totals**
- **Display of orders, mapped with their trade and order history details**
- **Display of trades**
- **Display of order transaction history**

The client uses an ultrafast technology that provides the user with real time data and notifications.

7.1 Configuration of limits and rules

The client provides the user with an easy and convenient interface to configure the order limits and rules for **Scrutiny**

Note: **Scrutiny** *will come with a default set of standard rules; however the user will be allowed to control what rules they wish to keep, what is removed and what is added. Version 1 will use the database management tools to maintain these limits and rules.*

7.2 Maintenance of limits and rules

After the initial configuration of limits and rules, the user can maintain them on a daily basis or as required via the client interface. Any change or addition during this maintenance would require both **Scrutiny** and the client to be restarted.

Note: Version 1 will use the database management tools to maintain these limits and rules.

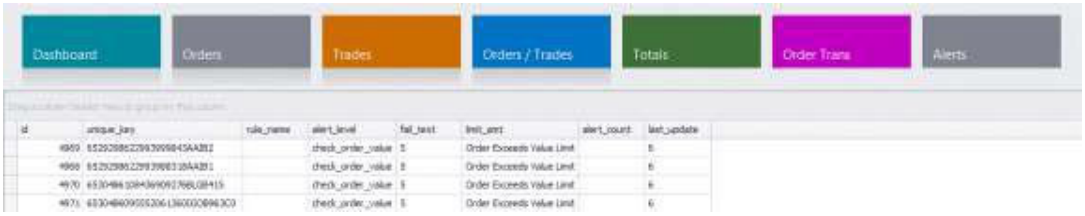
7.3 The dashboard

The client dash board display will provide two main components for the user to identify any irregular order movements.

- **Alerts Panel**
- **Order Totals**

7.3.1 Alerts Panel

The dashboard's alerts panel is designed to notify the user of any alerts triggered by **Scrutiny** in real time.



7.3.2 Order Totals

The dashboard's order totals panels display the current totals of the processed orders by the following categories.

- **Firm Totals**
- **Trader Totals**
- **Client Totals**
- **Security Totals**
- **Nested totals [Trader-Client] [Trader-Security] [Trader-Client-Security]**

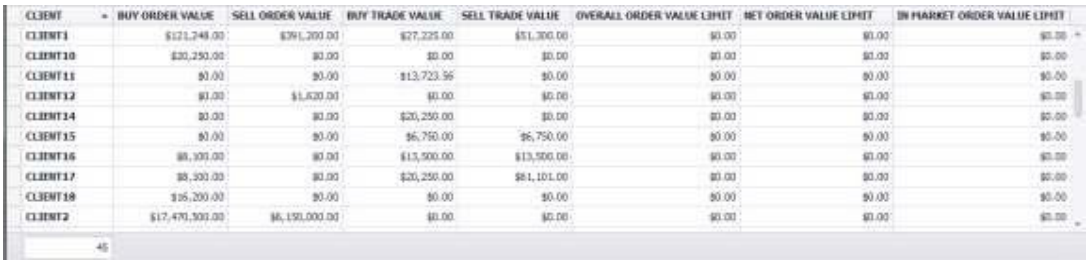
Trader Totals

This grid displays the following current Trader Totals:

At the bottom of the trader totals panel, the sum of them, i.e. firm's totals are displayed.

Client Totals

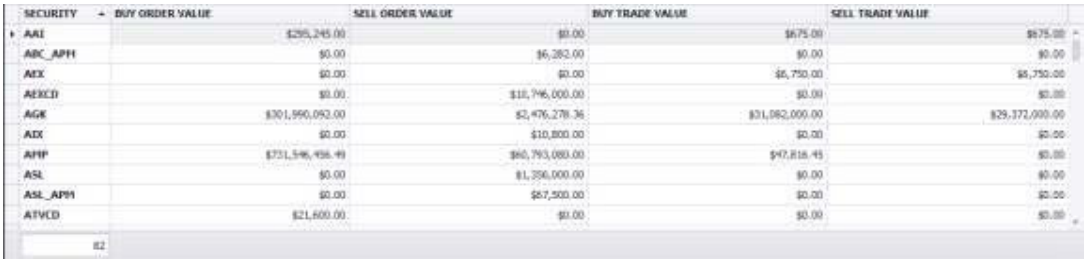
This grid displays the following current Client Totals:



CLIENT	BUY ORDER VALUE	SELL ORDER VALUE	BUY TRADE VALUE	SELL TRADE VALUE	OVERALL ORDER VALUE LIMIT	NET ORDER VALUE LIMIT	IN MARKET ORDER VALUE LIMIT
CLIENT1	$121,248.00	$391,200.00	$27,225.00	$51,300.00	$0.00	$0.00	$0.00
CLIENT10	$20,250.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
CLIENT11	$0.00	$0.00	$13,723.56	$0.00	$0.00	$0.00	$0.00
CLIENT12	$0.00	$1,620.00	$0.00	$0.00	$0.00	$0.00	$0.00
CLIENT14	$0.00	$0.00	$20,250.00	$0.00	$0.00	$0.00	$0.00
CLIENT15	$0.00	$0.00	$6,750.00	$6,750.00	$0.00	$0.00	$0.00
CLIENT16	$8,100.00	$0.00	$13,500.00	$13,500.00	$0.00	$0.00	$0.00
CLIENT17	$8,100.00	$0.00	$20,250.00	$81,101.00	$0.00	$0.00	$0.00
CLIENT18	$16,200.00	$0.00	$0.00	$0.00	$0.00	$0.00	$0.00
CLIENT2	$17,470,300.00	$6,192,000.00	$0.00	$0.00	$0.00	$0.00	$0.00
	45						

Security Totals

This grid displays the following current Security Totals:



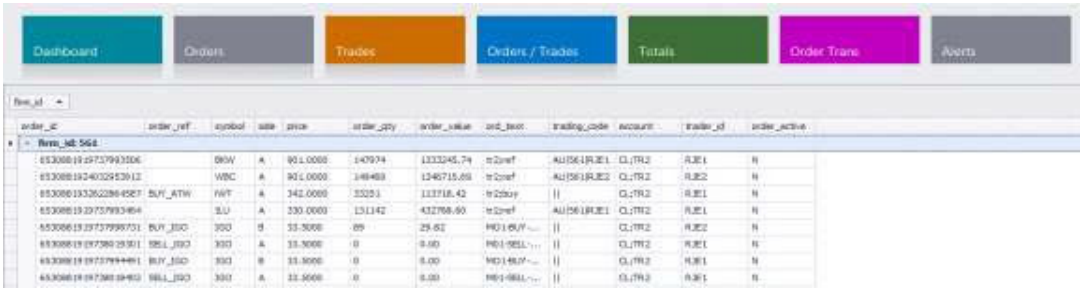
SECURITY	BUY ORDER VALUE	SELL ORDER VALUE	BUY TRADE VALUE	SELL TRADE VALUE
AAI	$295,245.00	$0.00	$675.00	$675.00
ABC_APH	$0.00	$6,282.00	$0.00	$0.00
AEX	$0.00	$0.00	$6,750.00	$6,750.00
AEXCD	$0.00	$18,796,000.00	$0.00	$0.00
AGK	$301,990,092.00	$2,476,278.36	$31,082,000.00	$29,372,000.00
AIX	$0.00	$10,800.00	$0.00	$0.00
APH	$731,546,456.49	$60,793,080.00	$47,816.45	$0.00
ASL	$0.00	$1,356,000.00	$0.00	$0.00
ASL_APH	$0.00	$67,500.00	$0.00	$0.00
ATVCD	$21,600.00	$0.00	$0.00	$0.00
	82			

7.4 The Display of Orders

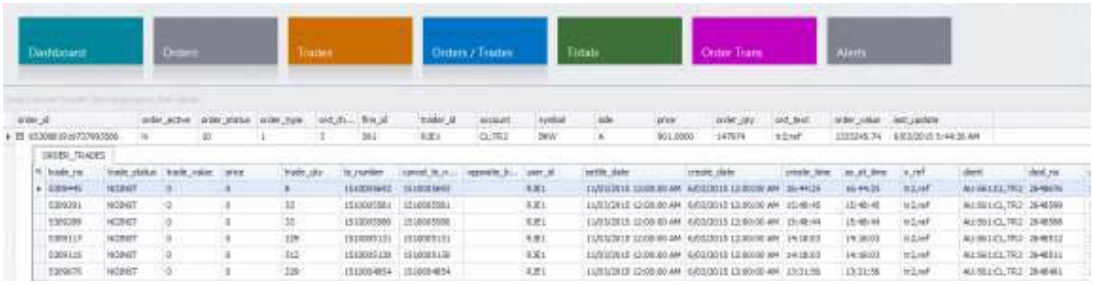
The client's display of orders in real time is continually updated with the latest order details. This grid data is able to be sorted, grouped and filtered by the user.

For the ease of use and convenience:

- The latest/latest modified orders are shown atop of the grid and the all orders are displayed in a chorological order of their time of creation/modification.
- The latest/latest modified orders are highlighted in the grid, allowing the user to easily identify them.

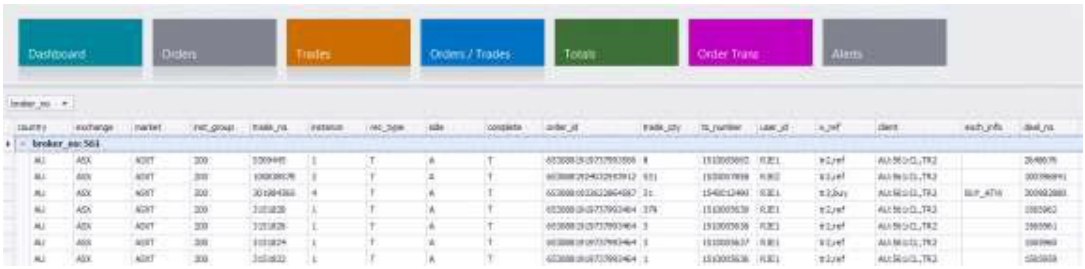


The orders in this grid are mapped with their trade and order history details and if a user clicks on a particular order, the trades related to the order and the order history/order transaction details of the order are displayed.



7.5 The Display of Trades

If a Trades feed is configured, the client provides a separate grid that shows all the trades of that day. This trade data is displayed in the same chronological order as the orders grid. This grid data is able to be sorted, grouped and filtered by the user.



7.6 The Display of Order Transactions

The client's display of order transactions is in their chorological order. This grid data is able to be sorted, grouped and filtered by the user.