FIX Trading Community – Microstructure Working Group

Recommendations for Algorithm Testing Test Cases

Background

The original draft of the MiFID regulatory technical standards (RTS) made some very specific statements regarding how algorithms should be tested and the types of scenarios that should be covered as part of such testing. The latter specifically made reference to definitions of 'disorderly trading conditions' and 'stressed market conditions' as follows:

RTS Text from Dec 2014 Draft	Disorderly Trading Conditions	Stressed Market Conditions
RTS 13 (requirements for investment firms)	• Price formation being significantly disrupted • Systems' performance is significantly affected by delays and interruptions • Multiple erroneous orders and/or transactions are experienced • Capacity of trading venues requires to be increased	• An increase or decrease in the number of messages being sent to and received from the systems of a trading venue • A significant short-term changes in terms of market volume • A significant short-term changes in terms of price (volatility) • An impairment of the performance of the trading systems of a trading venue or of the members and participants
RTS 14 (requirements for RMs, MTFs & OTFs)	• A trading system's performance which is significantly affected by delays and interruptions • Multiple erroneous orders and/or transactions, including cases where orders are not resting for sufficient time to be executed • A trading venue has insufficient capacity	• An increase or decrease in the number of messages being sent to and received from the systems of a trading venue • A significant short-term changes in terms of market volume • A significant short-term changes in terms of price (volatility)

The updated draft RTS as of September 2015 has removed the definition of disorderly and stressed market conditions and also removed the obligation for trading venues to ensure that algorithms are fully tested (this obligation now remains entirely with the investment firm). It would therefore appear to be a matter for each individual investment firm to determine how to test their algorithms and define the scope of the testing. This document seeks to provide guidance on this topic, establishing recommendations for the scope of testing and guidance on how to perform such testing. This has been prepared by the FIX Trading Community Microstructure Working Group which is comprised of industry experts from investment firms, trading venues and suppliers of technology products/services to both.

Though the original definitions of disorderly trading conditions and stressed market conditions are not in the latest draft RTS, the Working Group still feels they provide a good basis to define test cases. We categorise 'disorderly trading conditions' as market disruption events being caused by issues with an algorithm or trading venue system, e.g. capacity, delays, runaway algorithms. We categorise 'stressed market conditions' as where markets are undergoing unusual volatility/volume due to external forces (i.e. nothing to do with malfunctioning algorithms or trading platforms). A good example would be the Swiss franc revaluation in early 2015.

Overall Requirements

The MiFID RTS places the obligation to test algorithms on the investment firm operating the algorithms. Though it is ultimately up to individual investment firms to determine how best to achieve this, the following guidelines have been prepared to assist investment firms in meeting these requirements.

- Tests should be completely repeatable (i.e. same order book behaviour, same market data inputs etc.). This is to allow re-testing of an algorithm that fails a test and has since undergone remediation.
- Tests should also undergo some pseudo-randomisation to minimise the risk of overfitting algorithm behaviour to precise test scenarios. The test scenarios below provide some examples of how this could be used, e.g. randomising the timing of certain events in the test environment.
- This second bullet obviously contradicts the first bullet, the implication being that the test environment should be able to operate in both a fully repeatable mode and a pseudo-random mode, with only the latter being eligible for an algorithm to pass the tests.
- The test environment should be able to simulate various stressed market conditions, either by using historical replays of particular trading days or by having an order book simulator capable of replicating such behaviour.
- The test environment should be able to simulate other market participants, specifically those with algorithms themselves causing disorderly or stressed market conditions (the test scenarios below contain examples of this).
- The test environment should, where relevant, be able to simulate the behaviour of multiple order books trading the same instrument (e.g. primary market plus MTFs for equities) and simulate disorderly or stressed market conditions against either the market the algorithm is directly trading on, or some combination of one or more of this and the other markets.
- The test environment should be able to score the behaviour of the algorithm against agreed, predetermined and objective criteria (see 'success criteria' below) and log/retain this in a format suitable for the investment firm's compliance function. It is recommended that this log contain:
  - The date and time the test was run.
  - The identifier of the algorithm (the 'investment' or 'execution' id as referenced in the RTS on order record keeping and transaction reporting).
  - Pass/fail status together with the actual test scores for the algorithm as per the test criteria.

Criteria for Testing & Retesting Algorithms

The MiFID RTS states that investment terms shall 'set up clearly delineated development and testing methodologies' prior to the 'initial deployment or substantial update' of a trading algorithm. Though it is up to individual investment firms to interpret this, we make the following recommendations as regards good practice in this regard:

- Testing should be performed for each new algorithm.
- Testing should be performed for an existing algorithm which:
  - Is being used on a market that has a different market model or order book characteristics to the market(s) on it is currently used (e.g. an algorithm currently in use only in continuous trading sessions that has been adapted for use in auctions).

    - Is being used on a different asset class (e.g. an algorithm currently in use for equities being used for futures).
    - Has undergone a material change in functionality, for example (note this is not intended as an exhaustive list):
        - Addition of a new trading behaviour to an existing algorithm (e.g. adding dynamic volume participation logic to a previously static percent-of-volume algorithm).
        - A change in key input information (e.g. moving to a different market data infrastructure or provider).
        - A substantial change to the algorithm software code even if it does not result in any expected functionality changes (e.g. code refactoring, performance optimisations, code merges).
- It is not considered necessary that algorithms be retested based on infrastructure changes such as hardware or operating system version upgrades.

Success Criteria

An algorithm should be considered to have passed if the test scenarios are executed and none of the following occur:

- The algorithm does/would not trigger a circuit breaker (in a simulation environment it is sufficient to verify that the simulated market with the algorithm running does not undergo a significant price move compared with the simulated market running without the algorithm – 'significant' to be interpreted as being large enough to trigger an exchange circuit breaker)
- OTR remains below a certain limit
- The algorithm does not cause market impact (similar to the first point)
- The algorithm does not significantly increase participation in the market (which can be measured as being above an expected percent of volume)
- The algorithm, where appropriate, exits the market and, if so, does so without causing or exacerbating stressed market conditions

Test Scenarios

The table below describes sample test scenarios, test execution steps and associated test infrastructure requirements. Not all tests may be relevant for each algorithm or all market models/trading phases against which a particular algorithm may operate, and investment firms will need to use their judgement in this regard. It is recommended that appropriate tests from this table be performed for all trading phases (e.g. continuous trading, auctions) and market models (e.g. continuous trading, RFQ) that the algorithm is designed to support.

Requirement	Test Scenario	Test Method (including market data reqts)	Implementation details
Disorderly Trading Conditions			
Venue trading system performance significantly affected by delays	Trading platform slows down (e.g. due to high volumes), with 'slow down' meaning any of: • Latency in order acks • Latency in order cancels • Latency in fill distribution • Latency in market data distribution For multi-market algorithms, test both for: • One trading platform slowing down. • Multiple trading platforms slowing down.	Artificially, and on a randomised basis (both randomise the start and end time of the slowness, and the extent of the slowness) slow down the trading platform and run the algorithm against the slow platform. Three separate tests, targeting different aspects of the trading platform: • Order entry • Distribution of fills (if the platform handles these separately to order processing) • Market data For all three cases, we can consider 'latency' to be: • A multiple of (say 100x) the platform's typical latency (typically for European order driven markets this will be latency of around 10ms to 100ms). • A set delay of several milliseconds or more (depending on the type of trading platform). • Use production statistics from the actual trading venues and use 99th percentile figures.	For order entry latency, can introduce latency in gateway to order book (effect is to delay the order ack and result in missed fills). There is also a variant where the gateway is running at full speed but the order book itself is slow (where you have slow acks but still likely to get the fills you were expecting, those being delayed also). Should leave this choice to discretion of test system operator. For fill distribution latency, can do this either at order book level or gateway level (but leave to discretion of test system operator). Market data latency – slow down market data feed handler (while NOT slowing. Consider the scenarios where: • Order book is slow, others are fine • Order book is fine, gateways are fine, market data is slow • Order book is fine, gateways are

Requirement	Test Scenario	Test Method (including market data reqts)	Implementation details
			slow , market data is fine
Venue trading system performance significantly affected by interruptions	Trading platform disconnects participants • The participant doing the testing • Other participants Where the trading platform supports cancel-on-disconnect behaviour (and this is used in the live environment) then the testing needs to use/replicate such behaviour.	Run the platform (with regular latency) and, on an unpredictable (i.e. random-timed) basis, disconnect and reconnect a participant • Both the one doing the testing • Somebody else We are looking for both the disconnection and the reconnection/restart not to cause any problems (e.g. by accidentally replaying orders). • Disconnect and reconnect rapidly (i.e. around 1 second, depending on platform's capabilities) – this should test the algorithm's ability to 'recover' quickly • Disconnect and reconnect some time later (e.g. around 10 minutes, to be randomised) – to ensure that the algorithm logic does not cause disorderly conditions on restart e.g. by trying to unwind positions, catch up on trading etc.	Randomly disconnecting the testing participant • Test system to disconnect ALL participants at random times within an agreed time window. • One of each type of disconnection – short duration and long duration. Note 'disconnect' and 'reconnect' in practice may involve the test system disconnecting the participant and then being able to accept reconnections a) soon, b) much later (noting that in practice, many venues require participants to reconnect to them, rather than the other way round).
Multiple erroneous orders and/or transactions	We are interpreting this to mean an environment where one or more participants are repeatedly and rapidly placing orders and cancelling them (i.e. sending an order, cancelling it, then sending another order, cancelling that and so on). A second test is to perform a number of executions and then bust them (for markets which actually support this).	Requires a 'dummy participant' with such an algorithm. • Test participant algorithms without this running • Test again with this running Second test would involve the venue (on an unpredictable basis) busting a large number of executions.	Within an agreed time window, for a randomised period, the test system actives the 'dummy participant' and then stops it. Then, at a later randomised time, have the test system bust some of the executions from the dummy participant.

Requirement	Test Scenario	Test Method (including market data reqts)	Implementation details
			"Rapidly" will vary from platform to platform so we propose calibrating this as: • Order entry/cancellation at a rate such that passive orders remain on the book (before cancellation) for a period of time equivalent to the gateway-gateway latency of the market. • Randomised mixture of passive and aggressive orders (passive to test the cancel capability, aggressive to obtain fills to bust later). Have periods of passive-only, aggressive-only and mixed. • Messages to be sequenced as an order followed by its cancel.
Multiple orders that are resting for insufficient time to be executed		As above, except that the orders are immediately cancelled (and hence no fills)	Covered above
Capacity of trading venue requires to be increased (i.e. reaches a capacity limit, has insufficient capacity for current volumes)	Single market reaches a capacity limit and a) slows down, b) disconnects from participants. These are both covered above	Covered above.	Covered above.
Stressed Market Conditions			
An increase or decrease in the number of messages being sent to and received from the systems of a trading venue	Scenarios covering erroneous orders are covered above under 'Disorderly', so we are treating this as purely covering genuinely busy/quiet situations. A good example would be Swiss franc revaluation (for a busy day). Consider three types of scenario:	For an increase can replay a historical busy day (e.g. Swiss franc revaluation) or simulate data (either completely synthetic or by speeding up/merging data replays). For a decrease, recommend using simulated data or use historical data from a quiet day (e.g. the period between Christmas and New Year).	Need to consider separately an increase in market data (e.g. quote) volumes, and an increase in trade volumes. Three permutations: • Increase in market data and trades. • Increase in market data, trade volumes as normal.

Requirement	Test Scenario	Test Method (including market data reqts)	Implementation details
	• The entire day is busy/quiet. • The day starts 'normally' and suddenly goes quiet/busy for a prolonged period (e.g. rest of day, 1 hour). • The day starts 'normally' and suddenly goes quiet/busy for a short period (e.g. 10 seconds). Consider this happening on: • A single instrument • An entire market.	For the per-stock test, best to pick a stock (from a set list) at random. For the time-based tests, pick a time at random within a specified time range.	• Market data volumes normal, trade volumes increased. Simplest way to increase both is to speed up some real data. To separate them out will require data simulation as you will need to vary the 'matchability' of the orders. • We should define a 'normal' message rate to be, say, a full year average to be recalibrated every year. • For 'busy', take a random one of a sample of the 10 busiest days observed over the year and double it. • For 'quiet', take a random one of a sample of the 10 quietest days observed over the year and halve it. For the time considerations, this is easiest done with simulation (except for the 'all day' test which could be done also with real data).
Significant short-term changes in terms of market volume	Use test cases above.	Use test cases above (but ensure sufficient execution ratios to achieve high traded volume).	Covered above.
Significant short-term changes in terms of price (volatility)	Same permutations as above. Include also: • Changes that do not breach a venue's controls • Changes that do (e.g. cause a volatility interrupt)	Should be able to find some real-world examples for specific stocks. For a market-wide, can use the Swiss franc example (very little pan-market since financial crisis, Japan tsunami), IOB in 2014 (on back of Ukraine situation). And simulation options as per above. Randomisation tricky using real data so	

Requirement	Test Scenario	Test Method (including market data reqts)	Implementation details
		recommend simulation.	
Impairment of the performance of the trading systems of a trading venue or of the members and participants	Similar to the first item under 'Disorderly', so we are assuming that this requires us to combine the above 'stressed conditions' tests with the first in the lists under 'disorderly'.	Covered above.	Covered above.