Guide to the Development and Operation of Automated Trading Systems

MARCH 2015

On behalf of FIA, we are pleased to present a *Guide to the Development and Operation of Automated Trading Systems*.

Automated trading systems have become increasingly established within financial markets, and are used by a wide range of market participants from highly automated trading firms through to asset managers and pension funds that use trading tools provided by vendors and brokers. Recognizing the importance of ensuring that automated trading systems are designed and operated safely, the FIA Market Technology Division assembled a committee comprised of brokers, exchanges and principal traders to discuss and document current approaches in developing and operating such systems.

The Committee has drawn on previous FIA work on risk management, software development, change management, and post-trade processing. In addition, the Committee has referenced work published by regulators, international standards bodies and exchanges to determine the scope of the *Guide*, and to subsequently develop a consensus of current practices. To this effect, the *Guide* outlines current practices across a broad range of related subjects such as pre-trade risk management, post-trade analysis, co-location, disaster recovery/business continuity planning, software development, change management, testing, security, trading system operations, and documentation.

It is hoped that the *Guide* will form a basis for further work on standards across any asset class and global marketplace where automated trading takes place. Regulatory bodies and standards organizations can use the *Guide* to better understand current practices and implement approaches that are consistent across jurisdictions and asset classes. Market participants can use the *Guide* as a resource for developing, operating and reviewing automated trading systems within a comprehensive and detailed framework.

We expect that automated trading will continue to evolve as the industry evolves and FIA is committed to monitoring and supporting practices and procedures that improve the integrity and safety of the markets.

Yours truly,

Greg Wood
Leslie Sutphen
Co-Chairs, FIA Market Technology Division, Automated Trading Committee

CONTENTS

Introduction **5**
1 Pre-Trade Controls **7**
1.1 Maximum Order Size 8
1.2 Maximum Intraday Position 8
1.3 Market Data Reasonability 9
1.4 Price Tolerance 10
1.5 Repeated Automated Execution Limits 10
1.6 Exchange Dynamic Price Collar 10
1.7 Exchange Market Pauses 11
1.8 Exchange Message Programs 11
1.9 Message Throttles 12
1.10 Self-Match Prevention 13
1.11 Kill Switches 14
1.12 Cancel-on-Disconnect 15
1.13 Exchange-Provided Order Management 15
1.14 Identification of Automated Trading System Operators 16
2 Post-Trade Analysis **17**
2.1 Drop Copy Reconciliation 17
2.2 Post-Trade Credit Controls 19
2.3 Exchange Error Trade Policies 19
2.4 Audit Trail 20
3 Co-Location **20**
3.1 Fair and Equal Access 21
3.2 Network Infrastructure Equality 21
3.3 Measuring Latency of Service Provision 21
4 Disaster Recovery/Business Continuity **21**
4.1 Disaster Recovery 22
4.2 Business Continuity 22
5 Automated Trading System Development and Support **23**
5.1 Software Development 23
5.1.1 Feature/Requirement Gathering 23
5.1.2 Development and Testing Environment Design 24
5.1.3 Source Code Management 24

→

CONTENTS (CONTINUED)

5.1.4 Source Code Implementation 25
5.1.5 Risk Controls Implementation 25
5.1.6 Source Code Review 25
5.2 Testing 25
5.2.1 Unit Testing 26
5.2.2 Functional Testing 26
5.2.3 Non-Functional Testing 27
5.2.4 Exchange-Based Conformance Testing 27
5.2.5 Acceptance Testing 28
5.3 Change Management 28
5.3.1 Initiation 28
5.3.2 Validation 28
5.3.3 Authorization 28
5.3.4 Identification 28
5.3.5 Scheduling 29
5.3.6 Communication 29
5.3.7 Deployment 29
5.3.8 Trading Precautions 29
5.3.9 Post Deployment Verification 29
5.3.10 Completion 30
5.3.11 System Configuration 30
6 Security 30
6.1 Policies Regarding Security 31
6.2 User Screening and Education 31
6.3 Cybersecurity 31
6.4 Control and monitoring of use and access 32
6.5 Vendor Management. 32
7 Trading System Operations 32
7.1 System Monitoring, Failure Detection and Recovery 32
7.2 Emergency Notification Procedures 33
7.3 Risk Management 33
8 Documentation of Policies, Procedures, and Systems 34
8.1 Documentation 34
8.2 Documentation: Ownership 34
8.3 Documentation: Usage 34
8.4 Documentation: Sign-Off 35
Glossary of Terms 36

Introduction

As trading has become increasingly more automated over the past decade, market participants and regulators have focused more attention on how automated systems are designed, developed, and operated. Although automation provides many benefits to the marketplace, there have been several notable disruptions to the market due to technical issues and inadequate oversight. As a result of these and other smaller disruptions, legislators and regulators are spending considerable effort to understand how automated trading systems are being designed and operated and have begun to propose new regulations to provide guidance on how systems are to be implemented and operated in a manner that will not disrupt or manipulate the marketplace. Some specific initiatives include:

- The SEC's Regulation SCI (System Compliance and Integrity), proposed in March 2013 and adopted in November 2014
- The German High Frequency Trading Act in May 2013
- The CFTC's request for comment on its Concept Release on Risk Controls and System Safeguards for Automated Trading Environments in September 2013
- ESMA's consultation papers and discussion papers on Regulatory Technical Standards and Implementing Technical Standards in May 2014 followed by draft standards in December 2014

In the interest of promoting consistent practices and standards across regulatory jurisdictions and exchanges, FIA, FIA Principal Traders Group (FIA PTG) and FIA European Principal Traders Association (FIA EPTA)[1] have participated in a number of working groups to respond to proposed rules and to work with official standards bodies to formulate a *Guide* for the development and operation of automated trading systems. Previously, these FIA groups have provided several recommendations for pre-trade risk management controls and system development, notably:

- Market Access Risk Management Recommendations, FIA, April 2010
- Recommendations for Risk Controls for Trading Firms, FIA PTG, November 2010
- Software Development and Change Management Recommendations, FIA PTG and FIA EPTA, March 2012

[1] FIA is the leading trade organization for the futures, options and cleared swaps markets worldwide. FIA's membership includes clearing firms, exchanges, clearinghouses and trading firms from more than 25 countries as well as technology vendors, lawyers and other professionals serving the industry. FIA's mission is to support open, transparent and competitive markets, protect and enhance the integrity of the financial system, and promote high standards of professional conduct. As the principal members of derivatives clearinghouses worldwide, FIA's member firms play a critical role in the reduction of systemic risk in the global financial markets. FIA and its affiliates FIA Europe and FIA Asia make up the global alliance FIA Global, which seeks to address the common issues facing their collective memberships.

FIA PTG is an association of more than 20 firms that trade their own capital on exchanges in futures, options and equities markets worldwide. FIA PTG members engage in manual, automated, and hybrid methods of trading, and they are active in a wide variety of asset classes, including equities, fixed income, foreign exchange and commodities. FIA PTG member firms serve as a critical source of liquidity, allowing those who use the markets, including individual investors, to manage their risks and invest effectively. FIA PTG advocates for open access to markets, transparency, and data-driven policy.

FIA EPTA is an association of European principal traders formed in June 2011 under the auspices of the FIA. FIA EPTA members consist of 25 principal trading firms that provide significant amounts of liquidity to European regulated markets and multilateral trading facilities (MTFs). FIA EPTA members deal on own-account using proprietary capital and do not have clients or act as deposit takers in any form. FIA EPTA members are authorised by various EU national competent authorities and/or supervised by the regulated markets on which they trade, and are further subject to EU and national conduct of business regulation.

- Order Handling Risk Management Recommendations for Executing Brokers, FIA, March 2012
- Drop Copy Recommendations, FIA, September 2013
- FIA Response to the CFTC Concept Release on Risk Controls and System Safeguards for Automated Trading Environments, December 2013
- FIA Response to ESMA MIFID II/MiFIR Consultation Paper and Discussion Paper, August 2014

This paper is intended to make recommendations regarding the development and operation of automated trading systems, including the controls that can be used to manage risk at the automated trader, broker or exchange level.[2] The paper is intended to be asset-class agnostic but draws heavily on practices that have evolved within the futures markets. It is hoped that these recommendations will be considered in the development of regulations for any marketplace where automated trading systems are used. It is also the intention of this paper to provide guidance to global regulators in developing consistent and formal approaches to providing either regulations or guidance on implementing and operating these systems.

It is important to note that these suggestions will apply differently depending on the type of entity operating the automated systems (automated traders versus brokers versus exchanges), the size of the firm, the type of system being operated (trading system versus matching engine), the system's performance requirements, the types of instruments being trading (highly liquid and automated versus less liquid and more manual), and the types of algorithms being used. Care should be taken to avoid implementing overly prescriptive standards or rules that impose a one-size-fits-all approach to all entities. It should also be noted that practices may change over time based on changes to market conditions, to the trading approach, and to the instruments being traded, so a formal approach to reviewing and revising practices should be undertaken.

[2] This paper uses the term "automated trader" to describe any trading entity that uses an automated system including hedge funds, buy-side firms, trading firms, and brokers who deploy automated algorithms. The term "broker" includes futures commission merchants, other clearing firms, executing brokers and other financial intermediaries that provide access to an exchange. The term "exchange" is meant to apply to exchanges, marketplaces, and matching services that facilitate the automated execution of trades.

1 Pre-Trade Controls

Automated trading systems are now used by an increasingly large percentage of market participants including exchanges, trading firms, banks, hedgers, asset managers and pension funds, all of whom may develop their own systems or use systems provided by third parties such as vendors or brokers. All users and providers of automated trading systems have a responsibility to implement pre-trade risk controls appropriate to their role in the market, whether initiating the trade, routing the trade, executing the trade, or clearing the trade.

Localized pre-trade risk controls—not credit controls—should be the primary tools used to prevent inadvertent market activity due to unauthorized access, system failures, and errors. These controls can be implemented at various points in the execution order flow—at the automated trader, at the broker or at the exchange itself. Such localized controls use various approaches and act on a very granular level. Similar pre-trade controls may often exist at multiple points within the order flow and are used to mitigate risk from different viewpoints, for example, a maximum size order would be employed by:

- An automated trader to prevent it from submitting an order to the market that is larger than its risk tolerance.
- A broker to block a customer from submitting an order larger than the limits previously determined as part of a risk review exercise.
- An exchange or other trading venue at a product level as a secondary control to avoid inappropriately large orders that could affect price discovery within a market.

To maximize the effectiveness of a suite of risk controls, their designs should be principles-based and consideration should be given to the location where controls are implemented within the trading lifecycle. The specific implementation of these risk controls should not be prescribed by external regulatory bodies because market participants and exchanges are the best equipped to understand the performance of their systems, the unique needs of their markets and instruments, and the nuances associated with introducing new functionality to their systems. Any regulation or requirement for risk control that is overly prescriptive may fail to take into account the unique characteristics of the diverse market participants, exchanges, trading strategies, and instruments that exist today, thus adding to rather than reducing risk. Further, prescriptive requirements may quickly become obsolete as markets, technology, and trading strategies evolve.

Without prescribing specific risk control implementations, a degree of standardization may be achieved across market participants, regardless of their trading strategy, by implementing risk controls at the exchange. By doing so, participants are required to pass each of their orders through the exchange's risk controls prior to being submitted for execution. This provides a baseline of risk controls within the marketplace regardless of the type of access used or the type of market participant. Transparency on exchange-level rules and policies is an important consideration. Public disclosure of rules or policies on exchange websites or through other public forums is strongly encouraged. If policies are changed, prompt dissemination to market participants should occur.

This section of the document is intended to complement and build upon previous recommendations by providing an overview of pre-trade controls from the perspective of deploying and using an automated trading system. This section will list different types of pre-trade controls and make recommendations for where they should be deployed within the order flow, and by whom. The implementation and configuration of the controls set forth below should be reviewed on a regular basis and updated as necessary.

1.1 Maximum Order Size

Maximum order size sets the maximum quantity that is allowed to be submitted per order. These limits are commonly referred to as "fat-finger" limits. Errors may be prevented by rejecting the order in the case of a limit breach.

This risk control should be applied whenever a new order is submitted or an existing order is modified. Requiring each order to pass pre-trade order size checks can facilitate the entry of all orders into the market within parameters that protect the natural price discovery process from aberrant and accidental behavior, such as generating unintentionally large orders.

An appropriate maximum order size should be applied across different types of instruments appropriate to typical activity. For example, different maximum order sizes may be applied to orders on futures, options or spreads on the same underlying instrument. These limits may be adjusted for each instrument and each trading venue.

Systems should be designed to prevent orders from being placed in cases where no order size limits have been set for an instrument.

Such limits may be set at the trading application level. Depending on the type of market access, the broker providing access should also set limits within their own trading infrastructure for indirect access participants or using an exchange-provided tool for direct access participants, based on a review of the risk limits appropriate for the automated trader using the automated trading system.

Exchanges should also use a similar control applied to all participants at a product level to prevent accidental disruption of the market caused by orders that are too large for the liquidity of the marketplace; however, care should be taken that order size levels set at the exchange level are not too restrictive. Exchange mandated limits should be published so that market participants are aware of them before designing and configuring their systems.

1.2 Maximum Intraday Position

A maximum intraday position is the maximum long or short position that can be taken within a given system intraday. Errors may be prevented by rejecting the order in the case of a limit breach. Warnings may be employed when the limit is close to being breached.

When a new order is submitted or an existing order is modified both current positions and working orders should be evaluated to determine whether a breach of the limit could occur. It is important to include working orders such that limits would not be breached if that order is filled, even though it may not be immediately executable.

Such position limits may be considered simple pre-trade risk limits as opposed to credit limits since an accurate picture of start-of-day positions is difficult to derive in a timely fashion across multiple execution channels. It is important to note that not all systems can use this type of limit. This may be the case for automated trading operations that leverage more than one trading system, over-the-counter trading, or floor-based trading. Where implemented it should be considered a "speed-bump" to prevent accidental overtrading and, as such, should be employed with appropriate post-trade risk controls (see Section 2).

Maximum intraday position limits are generally set by instrument, by individual trader and/or for the whole automated trader, and may be set within the trading application or a separate risk management system that oversees activity across multiple trading strategies.

Depending on the type of market access, the broker providing access may also set limits within their own trading infrastructure for indirect access participants or use an exchange-provided tool for direct access participants, based on a review of the risk limits undertaken with the automated trader using the automated trading system.

Brokers should have processes for setting and amending limits, and limits should be set by the authorized person independent of the trader responsible for the automated trading system.

1.3 Market Data Reasonability

A market data reasonability check is a tool designed to control whether the data used to generate orders by an automated trading system is within acceptable boundaries. Errors may be prevented by having the existence of aberrant market data escalated to the supervisor or support team of the trading system and orders generated as a result of this data cancelled or rejected prior to submission to the market.

Trading and risk management systems should have such checks on incoming market data as well as on values generated using the market data. For example, automated trading systems should have controls that validate whether actionable data is reasonable based on a variety of factors that may include the time since the last update was received, previous price, bid/offer spread, or deviation from an average price. If there appears to be a deviation from what is expected, then an alert should be provided that market data may be stale, and any orders should be blocked while the deviation is investigated.

Both exchanges and commercial data providers should make efforts to disseminate accurate data. This is especially important at times of high volatility when the possibility of incorrectly disseminated data (including a data outage) could affect the ability of participants to manage their risk.

1.4 Price Tolerance

A price tolerance limit is the maximum amount an individual order's limit price may deviate from a reference price such as the instrument's current market price, and is typically applied on orders generated from an automated trading system before the order is sent to the exchange. Errors may be prevented by rejecting orders with limit prices placed outside the acceptable range. Price tolerance checks should be applied when a new order is submitted or when an existing order is modified. Requiring each order or amendment to pass price tolerance checks makes it more likely that all orders entered into the market are within parameters that protect the natural price discovery process from aberrant and accidental behavior, such as generating orders unintentionally far away from the current market price.

Price tolerance limits should be set at the trading application level. Depending on the type of market access, the broker entity providing access may also set limits within their own trading infrastructure for indirect access depending on the requirements of the participant and/or the exchange.

1.5 Repeated Automated Execution Limits

A repeated automated execution limit is the maximum number of times a strategy or identical order is filled and then re-enters the market without human intervention. After a configurable number of repeated executions, the strategy should be disabled until an authorized person re-enables it.

Due to the dependency on the type of strategy and on market conditions, these controls should be set by the automated trader, and not by the broker or the exchange. The appropriate limit will vary depending on the strategy in use and should be configured accordingly.

While it is the responsibility of the automated trader to detect incorrectly generated repeated executions, exchanges and brokers may also detect suspected repeated executions through their regular monitoring of market activity, and should attempt to contact the automated trading system operator regarding any action deemed appropriate to protect market integrity.

1.6 Exchange Dynamic Price Collar

A dynamic price collar, also called price banding, is the maximum amount a new trade price can deviate from a reference price such as the instrument's last trade price, and is typically used by an exchange as part of their error trade policy (see Section 2.3). Errors may be prevented by rejecting the order in the case of a limit breach.

Exchanges should use dynamic price collars at a market level. These controls apply to all participants to prevent accidental disruption of the market caused by orders that are entered too far from current market price. However, care should be taken that exchange-set price collars are not too restrictive and are based on accurate estimates of volatility and market conditions on a per instrument basis. Price collars should operate on all instruments and should be published so that market participants are aware of them before designing and configuring their systems.

By applying dynamic price collar functionality in their trading systems, exchanges can help protect against extreme, unjustified price movements and lessen the occurrence of trade busts or price adjustments. Price collars have been proven to minimize erroneous trading by controlling the range of execution prices and the integrity of trades cleared through the clearinghouse by dramatically reducing the chance that a trade may be deemed erroneous and subsequently busted or adjusted.

1.7 Exchange Market Pauses

Exchanges may choose to pause trading when market conditions indicate that price discovery may be suboptimal and pausing the market for a finite duration would allow for the re-establishment of the price discovery process in a fair and orderly manner. Typically, these pauses are incorporated in other types of functionality such as velocity logic or stop logic that detects when stop orders are being elected, causing a cascade effect. Such functionality will introduce a trading pause, giving participants an opportunity to respond before the market moves an excessive amount.

Market pauses help prevent errors by avoiding trades that may be considered out of range and hence may be subject to being busted or adjusted, affecting the risk management of the counterparties involved.

Volatility per se is not harmful and is part of the market's price discovery process. Market pauses—and the parameters around their use—should not be mandated on a broad-brush basis. Exchanges should base any implementation on the expected activity for a particular instrument, and clearly define the parameters that would invoke a market pause. When designing the criteria to trigger a market pause, it is important to acknowledge that a pause of any length may adversely affect the price discovery process and may dramatically reduce a market participant's ability to manage risk. As such, the policies that govern the use of these mechanisms should be established with the goal of keeping markets open as much as possible. This goal can be accomplished by allowing instruments to trade in a price range sufficiently large enough to allow the marketplace to naturally mitigate transitory liquidity gaps and by leveraging other appropriate pre-trade risk controls such as price collars (see Section 1.6) and order size limits (see Section 1.1) to prevent a single errant order from triggering a trading pause. These policies should be published and include the length of the pause and the manner in which information on the invocation of a pause is disseminated, although the exchange may also invoke a market pause at their discretion during exceptional circumstances, for example when experiencing technical issues. The exchange should notify the marketplace immediately by electronic message in the event of a pause.

If a trading pause must be triggered because of a fundamental breakdown in the price discovery process, it is important that the duration of the pause is limited so as to minimize any disruptions to the marketplace. As demonstrated on futures markets during the Flash Crash in 2010, even a momentary pause of trading afforded by this type of functionality can be enough time to provide an opportunity for market liquidity to be replenished.

1.8 Exchange Message Programs

Exchanges are in the best position to monitor a market participant's messaging practices to help safeguard the integrity of the market and the exchange platform.

The exchange should be responsible for setting messaging measures for each instrument based on many factors including the capacity and performance of its network and matching engine, the matching algorithm, and the unique characteristics of the financial instrument, particularly as it relates to liquidity. Messages can include orders, cancellations, modifications, and notifications of execution. Messaging measures should not be dynamic because market participants need to know what is expected of them. Details of exchange messaging programs should be transparent and publicly available.

It is reasonable for messaging programs for designated liquidity providers to be different from those for other market participants because designated liquidity providers are often required to quote two-sided markets in many products simultaneously, and an overly restrictive limit will inhibit their ability to perform their duties and properly manage the risk associated with those duties.

One example of a messaging program looks at a participant's order-to-trade ratio which compares the number of orders submitted to the executed quantity. For each instrument or instrument group, the acceptable threshold ratios are set by the exchange and publicly documented. It is important to note that this type of analysis is not done in real-time but after a trading session is complete. Also, it may not be possible to set meaningful order-to-trade ratios for newer products or products that trade infrequently.

1.9 Message Throttles

Message throttles are controls designed to prevent excessive messaging which could disrupt, slow down, or impede normal market activity.

There are a wide variety of approaches to message throttles that can be applied at various points in the order flow. Exchanges can establish controls at their gateways that monitor message rates and send warnings or reject messages from a participant when certain rates of messages per second are sustained. Such controls reduce inadvertent market activity by preventing high message rates that can stress the infrastructure at the automated trader, broker or exchange. High message rates generated by one automated trader can introduce slowness for other market participants (for example more messages than a market data feed can consume or disseminate), and can cause risk as participants cannot process the latest messages because they are still processing earlier ones. To avoid introducing undue risk into the marketplace, a message throttle should not ever reject an order cancellation request due to breached message rate limits. Exchanges that implement message throttles should publish their limits.

Exchange-based message throttles may be supplemented by message rate limits at the market participant or broker level. If an automated trader chooses to implement their own message rate limits, the limits must be flexible in order to address the market participant's unique and diverse risk management requirements. Brokers may choose to implement such controls to minimize disruption to execution services due to abnormal activity from a customer. They would also prevent any knock-on effect to post-trade services caused by capacity constraints from processing abnormal rates of pre-trade activity for indirect market participants. Brokers should be transparent to their customers regarding the reason for the additional control and the maximum message rate that can be supported by the broker. Finally, care should be exercised regarding any attempt to mandate the use of message throttles as it would be difficult to devise universal controls that are appropriate for all trading strategies and all financial instruments. Implementation of such controls should not be limited to

specific types of market participants as this is likely to distort fair and even access to the market, and may ultimately impact market integrity. For example, controls and their associated costs applied exclusively to designated liquidity providers or market makers could potentially discourage them from performing the critically important role of providing liquidity.

1.10 Self-Match Prevention

Self-match prevention is functionality designed to prevent a market participant from inadvertently trading with itself. For the purposes of considering such functionality, it is necessary to distinguish between three types of self-match trades that could occur on an exchange:

- **Wash Trades**: Intentional self-matches created with the intention to distort or manipulate the market, generally prohibited by rules and regulations.
- **Bona Fide and Allowable Self-Match Trades**: Buy and sell orders for accounts with common beneficial ownership that are independently initiated for legitimate and separate business purposes by independent decision makers and which coincidentally cross with each other in the competitive market.
- **Inadvertent Self-Matches that Occur on More than an Incidental Basis**: Orders submitted by the same person, automated trading system, trading team or related groups of traders are matched despite best efforts to avoid self-matching.

Market participants should have policies and procedures that prohibit wash trades and other forms of undesirable self-match trades. A variety of tools may be used to prevent inadvertent self-matches.

It is important to note that due to the diversity of trading operations and strategies, there is not a one-size-fits-all solution to self-match prevention. For example, a market participant that predominately acts as a liquidity provider may not want its resting quotes to prevent new hedge orders from being accepted for execution by the exchange. Similarly, a market participant that rests large limit orders for extended periods of time may not want those orders to be cancelled as a result of submitting a new, aggressing order to the exchange.

Exchanges should offer participants a selection of self-match tools to allow market participants to tailor self-match prevention to their individual needs by offering various options (e.g. cancel resting, cancel new, cancel both, and decrement order size) and various levels of granularity (e.g. firm level, group level, trader ID level, customer account level and strategy level). However, providing flexibility can increase the complexity of implementation, and an appropriate balance between flexibility and complexity should be found. It should be noted that certain levels may be combined or offered in conjunction with another level.

An important benefit of the exchange providing self-match prevention is consistency across market participants in terms of available functionality and cost impact. However, given the different requirements from market participants and different implementations at the exchange level, such controls should remain optional, and a decision by a market participant not to implement available functionality because it does not suit their business structure should not be construed as intent to bypass responsibilities regarding self-matches.

1.11 Kill Switches

A kill switch is a control that when activated immediately disables all trading activity for a particular participant or group of participants, typically preventing the ability to enter new orders and cancelling all working orders. It may also allow for risk-reducing orders while preventing risk-increasing orders. This can be considered an effective safeguard against situations such as an automated trader breaching limits defined by a broker, or erroneous trading activity that may be caused by an automated trading system malfunction or the generation of unintended orders released into the market.

Activation of a kill switch is based on a decision that such action protects market integrity or the financial integrity of the counterparties involved. Such a control may provide exchanges, brokers and automated traders with an immediate and effective way to remove or reduce risk. The conditions under which a kill switch may be used by an exchange or a broker should be clearly communicated to their counterparties.

However, kill switches should be considered just one of many different types of risk controls that comprise an effective suite of risk controls, and only invoked based on a qualitative decision taken as a last resort when other actions have failed or may not be feasible. In an environment that has adequate pre-trade risk controls at all appropriate focal points for the automated trader, broker and exchange, a kill switch may ultimately be considered redundant.

Automated traders are encouraged to build their own kill switch functionality into their trading applications where it is possible to implement it on a sufficiently granular level to identify individual trading systems. Such functionality may be separate from the trading application itself and can be operated both by the trader and by the person responsible for risk. When made available, this functionality should be in addition to and as a final backstop for the pre-trade risk functionality outlined above.

A broker may want to implement kill switch functionality for both its direct and indirect automated trader customers, although typically the revocation of customer trading access takes place through the broker's pre-trade risk controls whether implemented within its own infrastructure or using exchange-provided tools for direct access customers. Such a control should be granular enough to identify individual customers and/or trading systems as appropriate.

Where a broker has to rely on an exchange-provided risk management control—for example for a kill switch for a direct access participant—the exchange control should operate at a suitable level to control only that customer's order flow and should not be shared across customers. It is important to note that exchange risk management tools vary in how they are implemented based on how the exchange identifies trading sessions or operator IDs. Where an automated trader also has access to the exchange-provided control, the automated trader should not be able to override a kill switch invoked by the broker.

Where an exchange provides such a control, there should be a registration process and entitlement system that requires automated traders and brokers to specify which staff are authorized to use the functionality. The system itself should provide explicit warnings informing the authorized users of the consequences of activating the kill switch.

1.12 Cancel-On-Disconnect

Cancel-On-Disconnect (COD) is a service provided by exchanges that monitors for a loss of connectivity between a participant's trading session and the exchange's trading platform. If a loss of connection is detected, COD initiates a best-effort attempt to cancel all resting orders for the disconnected session. COD provides participants the safeguard of knowing that all working orders are cancelled at the exchange in the event that the automated trader loses its connection to the exchange.

COD functionality at the exchange should be optional, allowing automated traders to decide whether COD mitigates risk by cancelling orders in the event of a disconnection or adds to risk in such a situation.

It should be at the discretion of the exchange—i.e., the entity responsible for triggering COD functionality—to define what disconnection means. For example, it might involve detection of a network-level error or even loss of application-level heartbeats. What matters is that the exchange triggers COD when it has determined that a trading session has suffered an unexpected disconnect.

In terms of which orders ought to be cancelled upon disconnect, it should be considered that many automated traders maintain multiple trading sessions (i.e., connections) to an exchange, and order cancellation should be done at the granularity of an individual session so that all orders originating from the disconnecting session should be cancelled and those originating from other sessions should remain working on the exchange.

It is important to note that it is increasingly common for brokers to also employ COD for their connections to the exchange. This allows the broker to manage their risk across customers in the event of a loss of connection. As with an automated trader, the broker should also decide whether using COD in the event of an issue mitigates risk or increases risk, and customers should be advised accordingly.

Brokers also provide the ability for customers to route orders to an exchange—or multiple exchanges—through the broker's infrastructure. The broker should advise whether it is possible to pass through cancel requests to the exchange in the event of an unexpected disconnection by the customer from the broker's infrastructure. At present this is typically unsupported, and the customer would need to contact the broker to manually cancel any working orders.

1.13 Exchange-Provided Order Management

The ability to manage orders independently from the automated trading system is an important risk mitigation device. Errors may be prevented by cancelling any working orders in the event that there is a system failure.

Exchanges should provide an independent mechanism for viewing and cancelling working orders for a given session or user. Such functionality should be independent from the trading access that might be subject to disconnection or disruption, and may be used in conjunction with COD functionality (see Section 1.12), or in cases where COD is not provided. Such exchange-provided order cancellation and COD ought not to be viewed as alternative approaches; they are often complementary. Alternative order cancellation channels also allow a firm to proactively pull orders on behalf of trading sessions that they have themselves deemed in error.

It is important to note that in the event of a major network failure at the automated trader, alternative order management channels may also be impacted. At that point the only mechanical means by which orders can be removed is the exchange's COD capability.

Brokers providing exchange access to automated traders may also have access to the same alternative order management tools. In the event of a major system failure, authorized personnel at the broker may use this tool to confirm that orders have been cancelled and/or initiate the cancellation on behalf of the automated trader.

1.14 Identification of Automated Trading System Operators

Exchange audit trails should be designed such that the depth of information provided enables exchanges, brokers, automated traders and regulators to correctly identify market participants and analyze their behavior. Passing such information along with the trade information at the time of the order, or shortly afterward in the clearing process, can be an efficient and cost-effective way of identifying the source of trading.

Currently, among the information sent to an exchange, and thus included within the exchange's audit trail, are the following:

- A unique Operator ID, such as a FIX Tag 50 or Tag 116, which can be used to identify the firm, head trader, traders or systems administered under the head trader, as well as the contact information for the firm and head trader
- The clearing firm account, execution firm ID, and client order ID
- An exchange code
- A unique sequential number, date and time
- An identifier or other information indicating whether the order was generated manually or by automated means
- The type of message (e.g., new order, modify, cancel, execution, mass quote, quote request)
- On execution messages, an indicator as to whether the order is partially filled, completely filled, modified, rejected, expired or the trade cancelled
- On all cancel messages not triggered by an Order Cancel Request, an indicator of origin of cancellation
- For rejected messages, an indicator of the reason for the rejection
- The contract and maturity date, the type of order, a buy or sell indicator, and the number of contracts
- The limit price or stop price, if any
- The type of customer and whether it is for a customer or firm account

Regulations and exchange rules may allow for certain eligible account managers and others to give-up trades to other firms or accounts. Such give-ups must be done in accordance with the appropriate clearinghouse rules and policies.

2 Post-Trade Analysis

Although pre-trade controls are important components of a system to prevent inadvertent market activity or malfunction, creating complex pre-trade functionality is likely to impose heavy constraints on the efficient operation of automated systems. Accordingly a combination of post-trade controls, monitoring and data collection should be used in conjunction with pre-trade controls to watch for potential credit events or unintended trading. These post-trade functions will vary depending on the size and complexity of the automated trader and the variety of asset classes being traded. These post-trade controls will vary depending on the size and complexity of the automated trader and the variety of asset classes being traded.

2.1 Drop Copy Reconciliation

Drop copy is a report that details a participant's execution activity on a trading venue and is generated in as close to real-time as possible. Drop copy feeds are different from cleared trade feeds in that they (a) may contain additional information to aid a participant's risk management, such as order state changes, modifications, rejections and cancellations, and (b) are generated at the point of execution, rather than when the trade has been cleared. Currently the contents and method of delivery for drop copy feeds vary by trading venue. All participants may use drop copies for real-time trade reconciliation, including automated traders and brokers. This reconciliation process typically compares the information provided by a drop copy in real-time with the trade notifications received from production trading sessions. This comparison process allows firms to reconcile their electronic trading activity with an independent source of exchange-provided trade notifications. In the event a discrepancy is found, the responsible party may take action immediately to address trading risk, determine the cause of the discrepancy, and resolve any issues.

Market participants may also supplement their risk management process by using drop-copy functionality to consolidate multiple trading session reports into a single data feed. This consolidated data feed may then be used by operational staff to more efficiently monitor a participant's overall trading activity.

Drop copies should be available for all trading venues and products whenever technologically practicable. Exchanges should seek consistency in the format of drop-copy reports to assist in consolidation across exchanges. Trade reports and other information provided by drop copy should be disseminated to the market participant in real-time or as near real-time as technologically and operationally practicable. Updates provided by drop copy, or any other order and trade report, should include any necessary information required to identify the order described in the update and interpret the changes to that order. Additional details may be provided to increase the utility of the order and trade report.

Message Fields (based on the FIX Protocol)	
Session-related Messages	Logon BusinessMessage Reject
Session Details	SessionID
Order Details	ClOrdID (Or any unique customer order ID) SenderSubID (Or any unique trader ID) OriginalClOrderID OrderTimeStamp ExecutionReport (all types supported - Fill, Partial, Cancelled, Rejects, etc.) Side OrderType OrderPrice StopPrice (if applicable) TimeInForce ExpireDate (if applicable) ExpireTime (if applicable) MaxShowSize (if applicable) MinOrderQuantity (if applicable) EffectiveTime (if applicable)
Instrument Details	Instrument/Symbol MaturityMonthYear (if applicable) StrikePrice (if applicable) PutOrCallMarker (if applicable)
Booking Details	Account AccountType All fields related to fill assignment and clearing instructions (e.g., agent versus principal indicator)
Execution Report Details	OrderStatus RejectReason (if applicable) TradeDate ExchangeOrderID ExchangeExecutionID LastQuantity LastPrice CumulativeQuantity LeavesQuantity AveragePrice ExecutionTimeStamp MultiLegReportingType (if applicable)

Those details may include:

Miscellaneous Details	Currency (if applicable) QuoteID (if applicable) IOIID (if applicable) CoveredOrUncovered (if applicable) ManualOrAutomated (where applicable) CountryofOrigin (where applicable) Long/Short (if applicable) OpenOrClose (if applicable)

Both exchange drop copy and feeds from the broker clearing the trades are good sources of information to be reconciled with an automated trader's own system. A frequent reconciliation process where the firm balances its trading systems to drop copy or clearing information can serve as an early warning for potential problems and can help mitigate risks due to errors or malfunctions.

2.2 Post-Trade Credit Controls

Brokers that carry trades for an automated trader should establish post-trade credit limits that are appropriate for the market participant's capital base, clearing arrangements, trading style, experience, and risk tolerance. Credit limits should be determined by a broker's assessment of their customer's assets and history, and should be monitored across the customer's entire portfolio. Monitoring of customer credit limits includes their utilization of margin on positions carried by the broker, executed through the broker, those "given in" from other executing brokers, as well as the collateral posted in customers' accounts at the broker clearing the trades.

It is important to distinguish between pre-trade risk controls, which are designed to prevent trading systems from creating market disruptions—i.e. what is acceptable in terms of order size, number of orders and other checks outlined in Section 1—and credit controls, which are designed to prevent a credit event and are calculated on a post-trade basis.

Credit controls are a key feature of how a broker manages its exposure to its customers through the different types of market activity in which they participate, and as such need to be employed on a post-trade basis due to the diversity of information required to accurately calculate exposure where market participants have the ability to use multiple systems and/or multiple brokers to access the market. In such circumstances both automated traders and the broker clearing the trades may use drop copies and clearing system trade feeds on a near real-time basis, and should also maintain this data for historical review.

Automated traders and brokers may set daily position and/or loss limits by account as a form of credit control. These limits should be monitored and alerts generated at appropriate thresholds so that a discussion can occur between the broker and automated trader to decide what action should be taken to mitigate risk before the limit is breached. Such post-trade controls have both a quantitative and qualitative nature, and judgment should always be exercised before invoking certain controls, for example a kill switch (see Section 1.11).

2.3 Exchange Error Trade Policies

Error trade policies at exchanges should be transparent, deterministic, robust and clearly documented so that all participants understand the consequences of an error. Such policies are important for the protection of the clearing members as well as individual participants, and should be as consistent as possible across exchanges and clearinghouses.

Robust error trade policies serve to protect all market participants, including counterparties to trades that may be deemed erroneous. Where error trade policies are unclear or open to subjective analysis, it is possible that in attempting to reduce the risk of the party responsible for the error trade, the exchange may introduce risk to the counterparty, as well as other market participants, who may have acted in accordance with just and equitable principles of the marketplace. Exchanges should seek consistency in practices for the same types of financial instruments across markets where possible. Exchanges should also use pre-determined "no-bust" or "non-reviewable range" criteria as part of any error trade policy. Error trade policies should be publicly documented and reviewed on a regular basis.

The ultimate goal of any error trade policy should be to promote a marketplace where all trades stand as executed. To that point, exchange provided pre-trade controls (see Section 1) such as price collars and maximum order size controls can minimize the need to invoke an error trade policy. In instances where allowing trades to stand as executed is not possible, a price adjustment should be attempted instead of busting a trade. However, it is recognized that there may be exceptional circumstances when a trade has to be cancelled to retain market integrity. In both situations the affected party must report the error to the exchange within the prescribed reporting window. The exchange should notify the counterparties as well as the rest of the marketplace as quickly as possible of both price adjustments and trade cancellations to both allow continued price discovery to resume and to allow the counterparties and all impacted market participants to mitigate their risk as quickly as possible. Notification of any bust or adjustment should be sent in electronic format to all affected parties.

2.4 Audit Trail

Automated traders that use, or brokers that permit customers to use, automated trading systems should have in place a system to save order-related audit trail data for the time period specified in exchange and/or governmental rules and regulations. Audit trail data is typically stored in a data warehouse, and the entity should have the capability to report or extract that data into a standard format.

Brokers should assist regulators when customer activity is under inquiry by regulatory bodies. Further, if a broker has actual or constructive notice that a customer is violating regulations or the rules of an exchange or clearinghouse, it might be found to be in violation of exchange or clearinghouse rules if it fails to take appropriate action.

It should be noted that brokers may also be required to maintain a surveillance system in addition to maintaining an audit trail of their customer activity. Such surveillance systems may operate on a real-time basis or may be used on a post-trade basis to investigate activity that may require review.

Access to a surveillance system, in conjunction with an audit trail history, may help them efficiently respond to regulatory inquiries and meet regulatory reporting requirements. It can also allow brokers additional visibility needed to manage their own risk and make decisions on continued appropriateness of customer activity.

3 Co-Location

Co-location is the offering by an exchange or marketplace of data center space and network connectivity to its execution facilities. This is in contrast to proximity hosting which is when an end user finds third-party space that is located as close as possible to the execution facility's matching engine. Both co-location and proximity hosting facilities typically offer connectivity to telecommunication, redundant sources of power, cooling, remote support, and security.

Market participants may use co-location and proximity services to take advantage of the lowest possible latency of access to execution facilities as well as minimizing infrastructure footprints and providing disaster recovery.

In order to provide fair and equal access to execution facilities, it is recommended that co-location services be offered by the exchange or marketplace. This will prevent certain parties from obtaining better located facilities and blocking out other parties from having equal access. In cases, where exchange-provided co-location is not available, third-party proximity hosts should adopt practices to promote fairness and equal access.

3.1 Fair and Equal Access

Exchanges or marketplaces should make co-location services available to all market participants and third-party providers that wish to lease space in their data center on a fair and equal basis. Third-party service providers should be permitted to provide technology services from the co-location facility on the same basis as a market participant. Fees should be equitable, with charges being proportionate to the amount of infrastructure taken as opposed to membership status or other criteria. Finally, where infrastructure or access may be physically limited (e.g., fiber conduits, rooftop space) exchanges or providers should have policies in place providing for equitable use and allocation of these resources.

3.2 Network Infrastructure Equality

Exchanges or marketplaces should offer equidistant cabling connections between customer cabinets and the exchange or marketplace access points. This will provide all market participants the same distance to the access point regardless of location in the data center. Optimally, all cross-connects within the facility will be equidistant to ensure that location does not matter.

3.3 Measuring Latency of Service Provision

In order to assist end users in determining whether the provision of co-location or proximity hosting is offered in an equitable manner, the providers may provide actual statistics on the latency between where the end user accesses the hosting facility to where the provider connects to the exchange or to the execution facility. Measurement may be provided in an agreed standardized format that can be compared to that of other providers. The provider may provide statistics on average latency as well as longest latency and shortest latency.

4 Disaster Recovery/Business Continuity

Market participants should have crisis management procedures in place for managing automated trading software and operational failures. The ability to manage a crisis should not be inhibited by an overly prescriptive crisis management procedure. Instead these procedures should be designed by the market participant that intends to use them and should be commensurate with the type of business they are conducting. For example, a firm handling customer trades should consider the needs of the customers when developing a disaster recovery/business continuity (DR/BCP) plan whereas a firm trading exclusively for its own account will have different DR/BCP needs. Given the diversity of market participants that exists today it is infeasible, and potentially dangerous, to design overly prescriptive crisis management procedures for all participants.

Plans should address internal significant business disruptions that affect the firm's ability to do business such as a fire or system failure as well as significant external business disruptions such as the failure of an exchange, weather disaster, cybersecurity breach or terrorist attack.

4.1 Disaster Recovery

A disaster recovery (DR) program for automated trading should include: a review of the systems and data center vulnerabilities and threats; establishment of adequate contingency and disaster recovery plans; validation of the plans via exercises and tabletop reviews; performance of regular reviews of systems to check for compliance with the requirements; and performance of regular reviews by a responsible party to address any changes that need to be made. The formality of such a plan is dependent on regulatory requirements and the size and complexity of the organization.

Strategy: A DR program should establish and maintain emergency procedures, backup facilities, and a plan for disaster recovery that allows for the timely recovery and resumption of necessary operations and the fulfillment of the responsibilities and obligations of the entity. The DR program should also take into consideration the firm's essential service providers by analyzing the risks presented by critical third parties.

Plans: Firms should consider a DR plan that is appropriate for their business. Such plans should designate disaster response personnel and include all necessary contact details. As no two business operations or crisis events are the same, procedures should be flexible enough to allow responsible personnel to take into account the facts and circumstances of a particular event while deciding the necessary course of action to take in response to the event.

Disaster recovery plans must be developed for all IT infrastructure, applications and services that are deemed necessary by applicable regulation(s) and/or senior management to provide for acceptable restoration of business operations within an agreed-upon or mandated return to operation, following interruption to, or failure of affected infrastructure, applications or services.

Testing: Disaster recovery plans must be practiced or tested and updated to provide for their continuing effectiveness. In addition to regular testing, plan reviews should take place any time material changes occur which would introduce or change planning for recovery.

To the extent practicable, a DR program should be coordinated with the DR/BCP programs of the other market participants upon whom it depends, in a manner adequate to enable effective and safe resumption of activity following a disruption.

4.2 Business Continuity

Business continuity involves ensuring that the essential elements of trading continue in the face of disruptive external events, internal system and environment outages, and hardware and software failures.

To minimize the impact of certain types of disruptions, firms should consider the utility of standby systems for production infrastructure such as servers and network hardware in addition to key services such as trading applications and supporting services such as back office and even business e-mail continuity. Business continuity plans should be tested and participation in exchange-sponsored failover testing when available is encouraged.

In addition, some firms may wish to set up a parallel trading environment in a secondary location in the event that widespread disruption occurs at a location.

5 Automated Trading System Development and Support

Automated trading system development and support is defined as all activities that must take place to design, develop, deploy, and maintain automated trading systems. Due to the broad adoption of electronic and automated trading systems there is a wide and ever increasing variety of market participants and exchanges that should establish these procedures.

Development and support procedures should not be mandated in a prescriptive manner. Rather, market participants and exchanges should have the flexibility necessary to establish procedures that are appropriate and proportional to their operations.

A variety of software development and support methodologies exist, from those that better support a very small and organic environment to those that better support a highly structured, multiple team environment. Each market participant and exchange should employ a methodology that promotes efficient communication, generates maintainable source code, produces software that is implemented to specification and is easy to support.

When establishing development and support procedures, organizations may consider already existing methodologies. Similarly, they may adopt a methodology that is unique to their organization provided it satisfies the principles described below. Where an organization is composed of multiple independent groups responsible for automated trading system development and support, it may be appropriate for each group to have its own such procedures.

5.1 Software Development

Software development includes the writing, testing, and maintaining of the source code associated with automated trading systems.

Organizations that are reliant on and responsible for the development of trading software should have a process in place by which they can implement new source code or changes to existing source code without exposing themselves or the market to undue risk. There is a wide variety of automated trading related software that is responsible for the process of transacting on public markets, including but not limited to trading venues, trading systems, clearing systems, risk management systems, and research systems. Organizations should take into account the unique needs of the systems they develop when adopting and applying a software development process. Any such process for software development should address the following areas:

5.1.1 Feature/Requirement Gathering

Feature gathering is the process by which engineers and support staff responsible for automated trading systems collect the requirements for such systems prior to and during their design, implementation, and support.

Organizations should have a process that allows for any requirements to be accurately conveyed to engineers and support staff. When designing such a process, organizations should take into consideration team size, team structure, and communication mechanisms. For certain organizations, it may be best to have multiple independent feature gathering processes to address the unique needs of their various teams.

While gathering feature requirements for an automated trading system the following should be considered: functionality requirements, hardware requirements, network and connectivity requirements, redundancy requirements, hosting requirements, support requirements, and the system's dependency on external software and infrastructure components.

5.1.2 Development and Testing Environment Design

Development and testing environments are a set of integrated software and hardware components designed to mimic the production environment of the system in a manner sufficient to allow for the programming and testing of the system.

- **Environment Design:** Development and testing environments should be designed proportionally to the nature and context of the system. Development and testing systems may be encapsulated in a single environment or they may be separated in two independent environments. Such environments should contain at least the minimum set of components required to implement and test the system, as determined by the responsible parties. Any material changes to such environments should be communicated to those that may be impacted by the change.
- **Separation of Concerns:** Hardware and software components in such environments should be sufficiently separated from the production systems, both within the market participants' organization and externally at all times. No action conducted within such environments should result in a change or impact of any kind to production systems. Production environment systems should not be configured to refer to development or testing environment systems. Similarly, data traffic from development and testing environments to production environments should not be allowed.
- **Software Components:** The version of software components included in such environments should either be the new version of the system being implemented or tested or the same versions of the respective components currently in the production environment where the system will run—this also applies to proprietary and third party systems. Software components within such environments should be deployed and configured in a manner that sufficiently represents that of the respective production environment.
- **Hardware Components:** Hardware components used for such environments should have the same or sufficiently similar specifications as the components used in the respective production environments. Specifications of hardware components used in testing environments may differ from the specifications of components of the respective production environments provided that these deviations do not have an adverse impact on the system's intended behavior.

Exchanges should provide robust simulation environments that functionally replicate the production trading environment. These environments should be made available to market participants for testing purposes. Any trading activity that takes place in these environments should have no economic value and impart no risk to those that use the environment. Where practical, exchanges should provide two such environments for their participants—one that replicates the exchange's current production systems and one that represents any potential changes to be made to the exchange's production systems.

5.1.3 Source Code Management

Organizations should maintain a source code repository to manage source code access, preservation, and changes. The source code repository may be used to ascertain when software changes were made, who made the changes, and the nature of the changes.

An organization's source code repository should preserve source code in such a manner as to allow for the reproduction of any version of the software that was used within a production trading environment. The period of this preservation should conform to an organization's data retention policies.

5.1.4 Source Code Implementation

Source code implementation is the process by which source code is written. Code may be written in many different programming languages. Specific programming languages should not be prescribed for use as it may force software engineers to work in a language in which they are not comfortable thus introducing unnecessary risk. Rather, organizations should empower their software engineers to use programming languages with which they are sufficiently proficient. Prior to selecting a programming language, consideration should also be given to a language's ability to implement the necessary functionality as defined by those responsible for establishing system requirements.

Similarly, source code may be implemented in many different programming styles. A programming style may include formatting, variable naming conventions, and code architecture. Specific programming styles should not be prescribed. Rather, organizations should empower their software engineers to use programming styles with which they are comfortable provided that it is understandable by those responsible for the development of the system. Where the source code is not sufficient to convey the code's underlying functionality to the necessary parties, code comments may be used to provide supplementary details.

5.1.5 Risk Controls Implementation

Risk controls are an integral part of the development of any automated trading software. Please refer to Section 1 for a more detailed discussion of appropriate controls.

5.1.6 Source Code Review

Firms should have a process by which software engineers may have their source code reviewed when deemed necessary. Any such review may be performed formally or on an informal, ad-hoc basis. It may be performed concurrently with the implementation of the source code as is the case with "pair programming" or it may be performed on a post-implementation basis.

The goal of such reviews may be to confirm that the source code will work as intended, to provide mentorship, or to convey the specifics of newly implemented code. When undertaken, such reviews should be conducted by persons that have the contextual, business, and programming knowledge necessary to provide a meaningful and accurate review.

5.2 Testing

Testing pertains to the work done by organizations to confirm that their trading systems and environments function as designed and within acceptable parameters. Organizations should have a process for testing software and infrastructure before they are released to the production environment. Testing is a critical component of the software development and change management process as it aids in the prevention of system issues within the production trading environment. It is widely acknowledged that testing may not catch every system issue; therefore, the following general principles should apply:

- All testing should be appropriate and proportional to the change being made.
- Testing should be performed in an environment that sufficiently emulates that of the relevant production trading environment (see Section 5.1.2).

- Variables and initial conditions for tests should be selected such that the software system being tested behaves like it will when used in production.
- Testing may be performed manually by a person or executed in an automated manner by software systems designed to execute automated tests.
- Testing should verify that the system being tested is functioning as intended and within acceptable parameters.
- Testing of systems associated with a change but otherwise unchanged themselves may need to occur prior to the production use of the change in question.
- Those responsible for development, testing, and support of a system should be responsible for determining which tests must be completed successfully prior to the production use of that system.
- Failing to successfully complete a test deemed necessary for verifying that a system is working properly should be a sufficient condition for preventing the release of that system. In this case, that system should not be used in production until that test is successfully completed.
- As a system evolves it is possible that previously necessary tests become obsolete. Those tests should be removed from the testing plan for the system.
- Provided they are still relevant, previously designed tests should be reused for future software changes to enable the continued proper functioning of the system being changed.

Testing may not identify an issue within a system. A sound testing policy should be supplemented with appropriate risk control and support policies. A variety of effective testing methodologies exist and each firm should employ a suite of testing tools that suit the unique needs of their business and the change in question. Those testing methodologies are described below.

5.2.1 Unit Testing

Unit testing is a type of software testing in which discrete units of source code are tested to verify they work as desired. An individual unit test should be designed in such a way as to minimize the degree to which it tests aspects of a code base beyond the discrete unit of source code being tested. Unit tests may be configured to run automatically throughout the software development and building process.

5.2.2 Functional Testing

Functional testing is a type of testing that confirms a system behaves as specified. A typical system has a finite set of behaviors that it may exhibit. Functional testing primarily attempts to confirm these behaviors. Similarly, functional testing may be used to confirm that a system does not behave in an unintended manner. Such testing may be manually administered by a responsible person or automatically administered in a simulation environment by software systems designed to run such tests.

Types of functional testing to consider include but are not limited to:

- **Integration Testing:** A type of testing that confirms the system behaves as specified when its individual components are combined and tested as a group.
- **Regression Testing:** A type of testing that confirms that no bugs are introduced into an existing system as a result of changes being made.

5.2.3 Non-Functional Testing

Non-functional testing is a type of testing that confirms a system's non-functional requirements are met when its various components operate with one another and with external systems. Typically, the goal of non-functional testing is to confirm that a system performs as expected when faced with environmental changes and extreme events.

- Although non-functional tests may be administered manually by humans it is typically more effective to administer such tests in an automated manner in order to simulate extreme events. These tests may be run during the software development or building process.
- Non-functional tests should be designed to expose the system to events that are at least equal to what is expected within the production trading environment and possibly designed to expose the system to events that are more severe than would normally be expected.
- There are several types of non-functional tests that should be considered when testing a system. The responsible parties should implement whichever functional tests are deemed necessary for the system in question. Types of non-functional testing to consider include but are not limited to:
  - **Load Testing**: Load testing is a type of non-functional testing that identifies the limit of the system's ability to properly process external input and internal events. This is typically accomplished by testing the system with increasing amounts of external input and internal events until the system no longer performs as designed. Some types of input/events to consider when designing load tests are market data updates sent/received, trades sent/received, and orders sent/received.
  - **Stress Testing**: Stress testing is a type of non-functional testing that confirms that the system operates in an acceptable manner during periods of atypical amounts of external inputs and internal events. Typically stress testing is accomplished by subjecting the system to atypical loads over varying periods of time.
  - **Performance Testing**: Performance testing is a type of non-functional testing that confirms that the system operates in a timely manner. Typically performance testing is accomplished through measuring the time it takes for components of the system to perform their assigned tasks.
  - **Scalability Testing**: Scalability testing is a type of non-functional testing that confirms the system performs as designed as it is extended to support use by more users or it is deployed to more environments. Some types of scaling to consider when designing scalability testing include introducing more traders to a system, extending a system to trade additional products, and extending a system to trade on multiple exchanges.

5.2.4 Exchange-Based Conformance Testing

Exchange-based conformance testing is a type of testing that typically follows a script of tests designed and administered by an exchange to confirm that market participants' systems interact with an exchange's systems properly. By administering and performing such tests, exchanges can confirm that each market participant system exhibits a baseline level of functionality that has been deemed necessary for maintaining orderly markets.

- Exchanges should provide market participants with an environment that sufficiently mimics the production trading environment for conformance testing.
- Exchanges should require that conformance testing is performed whenever a market participant wishes to deploy a new exchange-facing software interface to the production environment. Follow-up conformance tests should be performed when material changes have been made to previously approved exchange-facing software interfaces.
- Market participants are responsible for initiating conformance testing whenever necessary.
- Exchanges should provide an appropriate series of tests for market participants to perform in conjunction with conformance testing. In the event that a market participant's system does not perform certain functions described within a conformance testing script, the exchange may grant a waiver for the associated tests.
- Exchanges should provide documentation to market participants to confirm the successful completion of conformance testing.

5.2.5 Acceptance Testing

Acceptance testing is a type of testing that confirms that a system meets a minimum set of criteria for use in production. Any type of test may be considered an acceptance test. In addition to already existing tests, supplemental tests may be introduced by the system's end user or another person familiar with the requirements of the system to act as further acceptance testing.

Typically, supplemental acceptance tests are designed or performed by someone familiar with the end user's expectations and use cases as they pertain to the system being tested. These supplemental tests are often administered manually to mimic how a human interacts with the system, but they may be administered automatically where deemed necessary.

5.3 Change Management

Along with appropriate software development and testing practices, it is important for organizations to establish change management procedures. A core component of the change management process is auditability. Organizations should establish procedures for communicating requirements, changes and functionality related to their systems. A historical written record of material changes to systems should be maintained in accordance with the organization's record retention policies. This written record should include when a change was made, who made the change, and the nature of the change.

In addition to ensuring all material changes to production systems are auditable, the following steps should be followed when changing production trading systems:

5.3.1 Initiation

Initiation is the process by which a change is determined to be necessary and planned. Every system change is initiated to meet a business, technical, or external requirement. The initiator of the change should identify the requirement(s) or nature of the change.

5.3.2 Validation

Each system change should be validated for correctness prior to deployment to the production environment. The validation process should include a proportional level of testing as described in other sections of this document.

5.3.3 Authorization

Prior to deployment, a planned change should be reviewed and subject to approval by a responsible party. The depth of the review performed should be proportional to the magnitude of the proposed change. The approval process may happen in various departments depending on the type of change being implemented.

5.3.4 Identification

Each software change that is deployed to the production trading environment should have a unique identifier (e.g., version number) that may be used to differentiate it from other versions of the software that have been previously deployed to production.

5.3.5 Scheduling

Prior to deployment, a planned change should be scheduled for release into the production environment. This schedule should be communicated to the necessary parties and should be considered along with any other planned changes as well as market-impacting events such as scheduled economic events and market hours.

In some cases there is a need for an emergency (or fast track) change. These may be done to resolve incidents that recently happened and are typically scheduled for immediate release. There should be a clearly understood process for bypassing the normal steps in the change management process and for determining when such a bypass is appropriate.

Exchanges should consider any potential impact to market participants, compliance systems, and reporting mechanisms during the change management process. If a proposed system change may have a material impact on their market participants, the exchange should take steps to clearly communicate the expected impact to their customers as well as its proposed release date. When deciding on a release date, exchanges should take into account the time needed by their market participants to make any software or operational changes necessary to properly account for the proposed release.

5.3.6 Communication

Trading and support staff materially affected by the change in question should be notified that the proposed change is taking place prior to initiating the change. They should also be notified when the change has been completed or if it has been determined that the change must be rolled back.

Communication of changes may take place in an informal and direct manner (e.g., in conversation or via email) or, if deemed necessary, in a formal, highly structured manner (e.g., published release notes or formal user education sessions). Regardless of the communication mechanism used, organizations should have the goal of communicating necessary information to the affected parties in a timely manner. When determining the proper communication mechanisms, organizations should consider the relationship of those implementing the change to those affected by the change, the scope of the change, and the potential risks to the system as a result of this change.

5.3.7 Deployment

Deployment is the act of releasing a change into the production environment. Depending on the nature of the change, it may be appropriate to deploy to the entire production environment at once or to deploy the change in phases to further mitigate risk and ease the reversion of the change if necessary.

5.3.8 Trading Precautions

Traders should take any steps deemed necessary to prevent undue risk to trading operations and the marketplace during the proposed release. These steps may include ceasing trading activity, hedging risk exposure, or using backup trading systems.

5.3.9 Post Deployment Verification

Post deployment verification is the act of verifying the deployed system change and the state of the production environment for accuracy. During the verification process the following steps should be considered:

- **Measured Usage:** Where reasonable, substantive changes to a trading system should be activated initially with appropriately restricted risk limits and access to markets until the change can be validated.
- **Technical Validation**: Support staff should confirm that the software is working as designed.
- **Trading Validation**: Trading staff should confirm that the software is working as designed.

5.3.10 Completion

A successful validation should result in the completion of the deployment. If the change cannot be validated a proper course of action may include:

- **Reversion**: The production environment is reverted to its prior stable state.
- **Disabling of Functionality**: The portions of the software change that cannot be validated are disabled until such time as they can be validated or a new software deployment may be released to production.

In the event that a change to software causes an incident or requires reversion, there should be a process for implementing this reversion and communicating the issue to the necessary parties for risk mitigation. It is also important to review the deployment, with regards to both what went according to plan and what did not. This "post mortem" process provides both an educational purpose for those involved in the deployment as well as a validation of approaches taken during deployment for those ultimately responsible for oversight of the automated trading system.

5.3.11 System Configuration

Depending on the design of the software and the needs of its users, system configuration can occur at the point of initial release or when the software is changed, or configuration elements can be changed independent of an overall software change.

Configuration elements can be fixed components of an overall software platform which require a full re-release when changed. More often, however, configuration components will be independent elements that can change on demand to support a more flexible need.

Typical configuration parameters for automated trading systems may include externally referenced hardware, databases, and other peripheral components. They may also include risk limits, algorithm parameters, and other elements that define system behavior.

When making changes to a system's configuration, consideration should be given to the impact of that change on the system's behavior. If deemed necessary, the system should be disabled prior to committing a configuration change to avoid potential unintended consequences associated with making a configuration change while the system is running.

6 Security

The software and hardware environments used to engage in automated trading may be located in remote locations such as data centers and offices. In addition, the automated trading systems may be connected by proprietary networks and may otherwise be isolated from broader internet access. However, there are still a number of controls and practices suggested for security of automated trading systems no matter how isolated a platform is from the public realm.

Each participant should take steps to limit the risk that its software access becomes lost or its controls disrupted, with potential market impact. In addition, each firm should take steps to protect its employees and users from inadvertent compromise of the firm's data in order to avoid harm to the participant and the marketplace. Finally, each firm should take steps to provide for the appropriate access to its development, testing, and production environments.

6.1 Policies Regarding Security

Each firm engaged in automated trading should have a security policy covering restrictions on access, controls required to prevent data breaches, delineation of who is responsible for security for each aspect of the environment, what sort of monitoring and logging is required, and what requirements should be made of vendors that provide components of the environment. The policy should be reviewed periodically.

6.2 User Screening and Education

Firms should verify that only appropriate users have access to their trading and development environments. Background and reference checks performed during the hiring process may be used for this purpose.

All users should be educated on how to maintain strong passwords, on how to maintain confidentiality and security practices, and on the appropriate use of devices in order to avoid loss of access or critical information. Users should be educated on how to avoid compromise of their access, how to support security initiatives, and how to report security incidents, should they occur.

6.3 Cybersecurity

Networks that facilitate the ability to communicate automated trading-related actions to a marketplace should receive the highest level of control and security. Access to these environments should be limited to users and administrative staff that have been educated on proper use, with a preference for use of more than one form of authentication before access is provided. The goal is to limit production devices from being accessed by unauthorized intruders, for example, if an authorized device is lost or compromised.

Password authentication should be required for access to all environments. The strength of a password is primarily improved by increasing its length. This may be supplemented by requiring that the password contain upper and lower case letters, numbers, and symbols. Passwords should be required to be changed on a regular basis and whenever a breach may have occurred.

Participants should implement up-to-date firewall and antivirus controls where deemed necessary in order to minimize or prevent an inadvertent or malicious compromise of any network which could be exposed to user changes or internet access. Internal networks used for trading should also have appropriate levels of authentication, separation, and logging of access.

Where possible, access to the internet or other insecure networks should be limited in highly sensitive areas such as trading environments. Where access to the internet is required, for example to obtain open source or vendor software, screening and staging of the software may be necessary before internal deployment.

6.4 Control and monitoring of use and access

To enforce security policies, a formal system for controlling and monitoring access should be put in place. Firms should consider physical security at their place(s) of business, co-location and/or proximity sites and be aware of the risk of access to their business infrastructure by unauthorized personnel.

Where feasible, firms should adopt measures such as electronic badges or other controls that limit physical access to their automated trading systems and/or management consoles at their place of business. In co-location and proximity sites, firms should understand the security measures provided by the facility and should adopt policies and procedures which, in conjunction with such measures, enhance overall security.

System access should be logged on each occasion. The extent to which logs are kept depends on the sensitivity of the environment and the time it might take to detect breaches and issues.

6.5 Vendor Management.

A written service level agreement should be put in place with each vendor of mission critical software, network infrastructure, and hardware, covering:

- Security of access to the firm's environment
- Confidentiality of the firm's information
- Warranty of security from breach of data, outside intrusion and disruption
- Continuity practices and service uptime warranties
- Attestation of screening and protection from viruses, malware, and other intrusions

7 Trading System Operations

It is important for automated traders, exchanges, and brokers to provide continuous monitoring of their automated trading systems. They should develop a set of practices for responding to trading disruptions and for mitigating financial losses as well as impact to the marketplace. Plans should be put in place to notify affected parties in the event of disruptions where necessary. Such monitoring and practices should be proportional to the size and the complexity of the automated trader or broker as well as the volatility and complexity of the marketplace being accessed.

7.1 System Monitoring, Failure Detection and Recovery

Automated trading systems should use a combination of monitoring tools and heartbeat detection to allow for timely discovery of system component and connectivity loss from exchanges. Heartbeat detection involves the monitoring of a persistent and repeated signal generated by hardware or software and an alerting system when such heartbeat either ceases or performs in an unexpected manner. Monitoring tools are generally more comprehensive than heartbeat detection and could detect such things as increased message counts, inconsistent messages, unexpected messages, and unauthorized access or intrusion.

Heartbeat detection may be sufficient for monitoring connectivity between external entities such as between an automated trader and an exchange. However, internal trading systems should be monitored for additional quantitative and qualitative aspects which may include excessive messaging, unexpected messages or traffic, excessive system resource utilization, interruption in the function of components, or errant behavior. Alerts should be generated and systems designed to either failover to parallel systems or discontinue activity in the case of disruptive or inappropriate functioning when deemed necessary.

Such monitoring, failover, and recovery processes should be understood by those operating and managing the systems and may be documented in more complex situations.

7.2 Emergency Notification Procedures

The following procedures serve to provide guidance in mobilizing an effective, efficient, and timely response to emergency situations. Undoubtedly, emergencies vary widely, so it is important to address a variety of market events and scenarios where action may be needed.

The following principles should be considered:

- Coordinate and provide prompt information to and from the exchange, broker, and automated trader.
- Maintain previously established Standard Operating Procedures (SOPs) to minimize miscommunications during an event.
- Evaluate the Emergency Notification Procedures and SOPs once the crisis resolution has been achieved.

It is the responsibility of exchanges, brokers, and automated traders, upon discovery of an emergency or disruption, to communicate the information to the affected trading community and regulators as necessary. In all cases, notifications should include event discovery, event progress, and event conclusion, with intermediate updates as necessary.

In addition to these overarching principles, the following *Guide*lines apply to each entity group:

- **The Exchange**: If an event is determined to affect a significant number of market participants and/or has the potential to require an emergency market halt, the exchange should send a notification message to the trading community and regulators. The exchange should circulate the information using multiple avenues of communication (e.g., text, email, website banner postings). The exchange should endeavor to distinguish these messages from normal business communications.
- **The Broker**: If an event is determined to have caused a significant disruption of a broker's business, the broker should notify customers, exchanges and/or regulators using previously established communication avenues (e.g., text, email, website banner postings).
- **Automated Trader**: If an automated trader suffers an event that has the potential to significantly disrupt the market or the broker's systems, the automated trader should contact the exchange or the broker, depending on the type of access. The notification should include information on whether there might be orders left in the market that need to be mitigated.

In the case of a disruption, it is important that automated traders design a process for communicating the disruption to their traders and advising them on appropriate action to take, which depending on the situation, may include cancelling orders or ceasing trading until the event is addressed.

7.3 Risk Management

Brokers who provide market access for customers using automated trading systems - which also includes the responsibility for monitoring and risk controls - should also have a formal risk management function independent from the traders to determine appropriate levels for pre-trade risk controls as well as to monitor the financial exposure of the traders. In the event that limits are approached or breached, the risk management function should have procedures for responding to such a breach in limits. Where possible, pre-trade risk controls (outlined in section 1) should be automated, requiring a manual override to enable a trader to continue trading. It is advisable to allow traders to undertake liquidation trades even when a limit may be breached.

Brokers who monitor and control customers operating automated trading systems through their market access should also have a formal risk management function independent from an execution desk but with responsibility for implementing pre-trade risk controls determined by the operational risk function of the firm and for taking action when credit limits are breached (e.g., cutting off access, communicating with customers, operating kill switches, and adjusting limits where appropriate).

8 Documentation of Policies, Procedures, and Systems

The above suggested practices for the development and operation of automated trading systems may rely on written and accessible descriptions of policies, procedures and system details associated with implementing them. The following are suggestions for documenting these practices.

8.1 Documentation

Documentation has historically been written text that may communicate policies, procedures, and system details. Newer forms of documentation include diagrams, video, photos and audio recordings. As long as the information is accessible and understandable, these newer, alternative forms of documentation may be used to represent the information. Although documentation may provide exhaustive details pertaining to its subject matter, it should not always be considered to be, or required to be, exhaustive.

Documentation may take several different forms, including but not limited to physical and electronic, to meet the needs of those using the document. Documentation may be established in a written language (such as plain English) or it may be established in other formats (such as source code or diagrams) provided the audience of the documentation can understand its contents.

8.2 Documentation: Ownership

Documentation should be owned by a member or members of staff with the necessary knowledge, expertise, and authority to validate the accuracy of the information within the document. A document pertaining to policies, procedures, trading systems, or trading environments may be created when it is deemed that there is a need to communicate such information in a written format. These documents should be updated whenever the owner of the document determines there has been a material change to the information contained within the document. Any such changes should be communicated to the relevant parties.

Documentation should be reviewed, as appropriate, by the document's owners to confirm the accuracy of the information it represents. The frequency of this review can be a function of regulatory requirements or criticality of the documented information. During this review, if it is determined that the document is no longer valid, action may be taken to stop referencing the document as current.

Documents should be stored in a manner that provides ease of discoverability and access to those that need them while preserving the intellectual property it represents. Organizations should take into account if the document is meant for internal or public use when establishing a storage policy.

8.3 Documentation: Usage

Documentation can be costly and efforts to create and maintain documents should be appropriately scaled to the size and scope of the activity. Determining whether documentation should or may be produced must take into account the overall benefits versus the potential risk of intellectual property theft, liability, misinterpretation and use for unintended purposes.

It is possible that an overall risk assessment may yield different answers depending on each firm's unique circumstances. Documentation can be useful whenever automated traders, brokers, and exchanges determine the need to reproduce events, to provide reference material, to inform large groups of people of operating procedures, and when it can be helpful to provide descriptions of scope and requirements to disparate groups. The format of documentation should be consistent with requirements set forth by those requesting the documentation.

Some aspects of automated trading systems and support to consider for documentation may include but not be limited to:

- **Regulatory Requirements:** Organizations are required, by regulators, to document certain policies, procedures, and actions.
- **Audit Trails**: Information regarding the actions taken by an automated trading system.
- **System Requirements:** An automated trading system's features and functionality requirements as defined by the business owners or end users of a system.
- **Project Work:** The work done to design, implement, test, release, and support trading systems and environments.
- **Systems Design and Functionality:** How automated trading systems and environments are designed and function when used.
- **Trading Systems Access Authorization:** Who has access to automated trading systems for order submission and cancellation purposes. Organizations should establish documentation to identify who has been granted access to which trading systems for trading.
- **Support and Operational Procedures:** The policies and procedures an organization follows to manage automated trading operations and support their automated trading systems and environments. These may include establishing who is responsible for the management and support of automated trading systems and environments, establishing a business continuity plan, setting and controlling risk limits, and establishing procedures to follow in the event of a trading error.
- **Change Management:** The process an organization follows to implement and release a change to their trading systems and environments.

8.4 Documentation: Sign-Off

Documentation sign-off pertains to a policy that establishes if a document needs to be formally acknowledged by a responsible party as accurate and having met the necessary governing criteria. Organizations should establish which documents require formal sign-off and when that sign-off is required. If deemed necessary, all sign-offs should include the responsible party's name, title, signature (physical or electronic), and date. Responsible parties may include system owner, system architect, business owner, business analyst, software engineer, quality assurance manager, or senior management.

GLOSSARY OF TERMS

A

Acceptance Testing
Acceptance testing is a type of testing in which the software is tested by an individual familiar with the purposes of the software to verify conformance of a system to the stated business requirements. Acceptance testing should be done in an environment that adequately represents the environment in which the software will be released.

Aggressing Order
An aggressing (or aggressor or aggressive) order is one that is marketable and can be immediately matched when it is received by the exchange matching engine against a passive order resting in the central limit order book.

Algorithm
The term "algorithm" broadly refers to a step-by-step procedure used for calculation or analysis. A wide range of computer programs—not limited to automated trading systems—are often made up of many algorithmic steps, often shared across multiple programs within the same organization.

Applications
An application is software that can be considered to meet a specific business requirement.

Audit Trail
An audit trail is a record of transactions that would typically identify information about the initiation of a transaction, the brokers participating in each transaction, the firms clearing the transaction, the terms and time or sequence of the transaction, as well as transaction receipt and execution time, and—when applicable—the customers involved.

Automated Trader
Automated Trader refers to any trading entity that uses an automated system including hedge funds, buy-side firms, trading firms, and brokers who deploy automated algorithms.

B

Bid/Offer
Bid/Offer refers to the prices displayed that represent the levels at which a financial instrument can be bought or sold. The bid/offer can be quoted directly, or derived from resting orders in a central limit order book, depending on the type of market. The spread (difference) between the bid/offer is an indicator of liquidity.

Broker
The term "broker" includes futures commission merchants, other clearing firms, executing brokers and other financial intermediaries that provide access to an exchange.

C

Cancel-On-Disconnect
Cancel-on-disconnect (COD) is a service provided by exchanges that monitors for a loss of connectivity between a participant's trading session and the exchange's trading platform. If a loss of connection is detected, COD initiates a best-effort attempt to cancel all resting futures and options orders for the disconnected session.

Central Limit Order Book
The central limit order book (CLOB) is provided by the exchange as a mechanism for price discovery. Orders can be placed at various price levels and the exchange matching engine will execute trades based on the appropriate algorithm for the market, for example Price/Time Priority, Pro Rata Trade Allocation or Batched Order Processing.

CFTC
Commodity Futures Trading Commission

Change Management
Change management refers to procedures for communicating, implementing, and tracking requirements and changes to functionality related to systems.

Cleared Trade Feeds
A cleared trade feed is a data feed that contains information about executed trades that is produced as a result of the clearing process. Such a feed may be in less than real-time and will not include information on unfilled orders or order cancellations. It may also include information on post-trade account allocation.

Co-Location
Co-location is the offering by an exchange or marketplace of data center space and network connectivity to its execution facilities.

Cross-Connect
A cross-connect is a network connection within a co-location or proximity hosting facility that connects a market participant's infrastructure to the exchange's infrastructure.

Cybersecurity
Cybersecurity refers to steps taken to limit trading and development environments from being accessed by unauthorized intruders or jeopardized from the introduction of unwanted software.

D

Direct Participant
A direct participant is market participant characterized by use of an automated trading system directly connected to an exchange without using a broker's infrastructure to route orders.

Documentation
Documentation is recorded information that may communicate policies, procedures, and system details.

DR/BCP
Disaster Recovery/Business Continuity Planning.

Drop Copies
Drop copy is a report that summarizes a participant's execution activity on a trading venue and is generated in as close to real-time as possible. Drop copy feeds are different from cleared trade feeds in that they (a) may contain additional information to aid a participant's risk management, such as order state changes, modifications, rejections and cancellations, and (b) are generated at the point of execution, rather than when the trade has been cleared.

E

Error Trade Rules or Policies
Error trade rules or policies are exchange or other market center rules or policies that describe the conditions under which trades that are executed at prices inconsistent with prevailing market conditions, typically due to an error, can be reviewed, adjusted, or cancelled (i.e., "busted") after execution. Such rules or polices also typically describe limitations to after–the-fact changes, typically in the form of "no-bust" or "non-reviewable" price ranges.

ESMA
European Securities and Markets Authority

Exchange
The term "exchange" is meant to apply to exchanges, marketplaces, and matching services that facilitate the automated execution of trades.

Exchange-Based Conformance Testing
Exchange-based conformance testing is a type of testing that typically follows a script of tests designed and administered by an exchange to confirm that market participants' systems interact with an exchange's systems properly.

Exchange Dynamic Price Collar
A price collar is a system safeguard aimed at preventing errors in order entry. A price collar determines a range around current prices within the central limit order book such that trades cannot occur outside of that range. For example, a price collar could be set where a trade cannot occur at a price level that differs by more than 10 percent from last trade price.

Exchange Error Trade Policies
See Error Trade Policies

Exchange Market Pauses
Exchange market pauses are temporary pauses in trading used by exchanges to prevent or limit market disruptions.

Exchange Message Policies
Exchange Message policies are policies for messaging measures for each instrument based on many factors including the capacity and performance of its network and matching engine, the matching algorithm, and the unique characteristics of the financial instrument, particularly as it relates to liquidity.

Exchange-Provided Order Management Tool
An exchange-provided order management tool is an independent mechanism provided by the exchange for viewing and cancelling working orders for a given session or user. Such functionality is independent from the trading access that might be subject to disconnection or disruption.

Exchange Simulation Environments
Exchange simulation environments are environments that are designed to mimic the production environment to enable conformance testing of new and modified trading systems.

F

Fat-Finger
The term "fat finger" describes a type of trading error caused by mistyping on a computer keyboard. The term has come to capture more generally any trading error caused by simple human error.

FIA
FIA is the leading trade organization for the futures, options and cleared swaps markets worldwide. FIA's membership includes clearing firms, exchanges, clearinghouses and trading firms from more than 25 countries as well as technology vendors, lawyers and other professionals serving the industry. FIA's mission is to support open, transparent and competitive markets, protect and enhance the integrity of the financial system, and promote high standards of professional conduct. As the principal members of derivatives clearinghouses worldwide, FIA's member firms play a critical role in the reduction of systemic risk in the global financial markets. FIA and its affiliates FIA Europe and FIA Asia make up the global alliance FIA Global, which seeks to address the common issues facing their collective memberships.

FIA European Principal Traders Association (FIA EPTA)
FIA EPTA is an association of European principal traders formed in June 2011 under the auspices of the FIA. FIA EPTA members consist of 25 principal trading firms that provide significant amounts of liquidity to European regulated markets and multilateral trading facilities (MTFs). FIA EPTA members deal on own-account using proprietary capital and do not have clients or act as deposit takers in any form. FIA EPTA members are authorised by various EU national competent authorities and/or supervised by the regulated markets on which they trade, and are further subject to EU and national conduct of business regulation.

FIA Principal Traders Group (FIA PTG)
FIA PTG is an association of more than 20 firms that trade their own capital on exchanges in futures, options and equities markets worldwide. FIA PTG members engage in manual, automated, and hybrid methods of trading, and they are active in a wide variety of asset classes, including equities, fixed income, foreign exchange and commodities. FIA PTG member firms serve as a critical source of liquidity, allowing those who use the markets, including individual investors, to manage their risks and invest effectively. FIA PTG advocates for open access to markets, transparency, and data-driven policy.

FIX Protocol
FIX stands for Financial Information eXchange. FIX is an industry standard for exchanging messages for financial instruments. The FIX protocol is commonly used for routing orders between participants and brokers.

Flash Crash
The Flash Crash refers to the sudden drop and immediate rebound in futures and securities prices that occurred shortly after 2:30 p.m. Eastern Standard Time on May 6, 2010.

Functional Testing
Functional testing is a type of testing in which well-defined software modules are combined to have their functionality tested as a group. Two types of functional testing that may be considered are "integration" and "regression" testing.

G

Give Up/Give In
Give-up/give-in refers to a trade that is executed through one FCM and cleared with another. The executing broker "gives up" the trade, while the clearing broker receives the trade as a "give in."

Granularity
Granularity describes firm level, group level, trader ID level, customer account level and strategy level.

I

Indirect Participant
An indirect participant is a market participant characterized by the use of an automated trading system that routes orders through a broker's infrastructure before they are sent to the exchange.

Infrastructure
Infrastructure is the hardware, network, or associated operating software used in conjunction with an automated trading system.

Integration Testing
Integration testing is a type of functional testing that confirms that the system behaves as specified when its individual components are combined and tested as a group.

K

Kill Switch

A kill switch is a control that when activated immediately disables all trading activity for a particular participant or group of participants, typically preventing the ability to enter new orders and cancelling all working orders. It may also allow for risk-reducing orders while preventing risk-increasing orders.

L

Latency

Latency is a natural delay in a system due to the time it takes to process and disseminate information.

Liquidity

Liquidity is a market attribute that describes the degree to which a financial instrument can be bought or sold in the market without affecting the price for that financial instrument.

Liquidity Provider

A liquidity provider is a type of professional trader whose orders more often than not supply liquidity to the market instead of demanding it. Liquidity providers typically perform a market-making function.

Load Testing

Load testing is a type of non-functional testing that identifies the limit of the system's ability to properly process external input and internal events.

M

Mass Quote

A mass quote is a function within some exchange systems that allows authorized customer systems to submit Mass Quotes messages to generate bid/ask pairs and two-sided markets for multiple instruments.

Market Data Reasonability Checks

A market data reasonability check is a tool designed to control whether the data used to generate orders by an automated trading system is within acceptable boundaries.

Market Participant

Market participant refers to any party participating in a market including automated traders, brokers, and end users.

Matching Engine

The matching engine refers to the allocation algorithm embedded in an exchange's computers to match marketable buy and sell orders within the central limit order book and convert them into executed trades. Several types of matching algorithms exist, for example Price/Time Priority, Pro Rata Trade Allocation and Batched Order Processing, and are chosen by the exchange on a product-by-product basis to match the requirements of the financial instrument and its participants.

Messages
Messages are instructions sent to and received from the exchange including orders, cancellations, modifications, and notifications of execution.

Message Fields
Message fields refer to fields within messages that are either sent to the exchange or received back from the exchange that contain discrete pieces of information. Message fields are defined for order transmission as well as drop-copy and clearing feeds.

Message Throttles
Throttles on message traffic and trade executions are controls that limit the number of orders (and cancellation or revision of orders) submitted and the number of trades executed.

Message Rate Policies
See Exchange Message Rate Policies

N

No Bust Range
A no bust range is a range of market prices within which trades executed at those prices will not be cancelled or "busted." See Error Trade Policies.

Non-Functional Testing
Non-functional testing refers to a type of testing in which well-defined software modules are combined to have their non-functional aspects tested as a group. Such non-functional aspects might include scalability, performance, stability, and usability.

O

Operator ID
The operator ID is a unique code that identifies the party or parties that entered or caused the entry of an order into an exchange system. The operator ID is included as part of each order message sent to the exchange and maintained in the exchange's audit trail.

Order Cancel Request
Order cancel request refers to a message sent to the exchange's matching engine requesting that a previously submitted order be cancelled and that a confirmation of cancellation be sent.

Order-to-Trade Ratio
An order-to-trade ratio compares the number of orders submitted to the executed quantity.

Order Types
An order type is an instruction that an exchange provides to participants to allow different interaction with the central limit order book. For example, a "market order" is an order to buy or sell that is to be executed at the best price currently available, and may trade at several price levels within the order book to be fully executed. A "limit order" is an order to buy or sell that cannot trade beyond its limit price.

P

Performance Testing
Performance testing is a type of non-functional testing that confirms that the system operates in a timely manner.

Post-Trade Credit Controls
Post-trade credit controls are limits set by a broker to manage its financial exposure to its customers through the different types of market activity in which they participate.

Pre-Trade Maximum Intraday Position
Pre-trade maximum intraday position is the maximum long or short position that can be taken within a given system intraday.

Pre-Trade Maximum Order Size Controls
Pre-trade maximum order size is a pre-trade risk control set at the automated trader level, the broker level, or exchange level (or all of the above) that sets limits on the size of an order submitted to the exchange's matching engine.

Pre-Trade Risk Controls
Pre-trade risk controls are controls used to prevent inadvertent market activity due to unauthorized access, system failures, and errors.

Price Adjustment
Price adjustment is a change in the price of an executed trade made in accordance with an exchange's error trade rule or policy.

Price Collar
See Exchange Dynamic Price Collars

Price/Time Priority Allocation
Price/time priority allocation is an exchange matching engine algorithm that fills buy and sell orders according to price and time priority, also known as "first-in-first-out" (FIFO). An incoming order's quantity immediately matches against each resting order at the same price within the central limit order book queue, decrementing each resting order based on its position within the queue. Resting orders at the same price level are given matching priority based on the time they arrive at the exchange with the oldest order having the highest priority.

Price Tolerance Limit
A price tolerance limit is the maximum amount an individual order's limit price may deviate from a reference price such as the instrument's current market price, and is typically applied on orders generated from an automated trading system before the order is sent to the exchange.

Pro Rata Trade Allocation
Pro rata trade allocation is a matching engine continuous algorithm implemented by exchanges that will fill orders according to price, order size and time within the central limit order book. An aggressing order's quantity is multiplied by each resting order's pro-rated percentage to calculate allocated trade quantity. An order's pro-rata percentage is calculated by taking order quantity divided by total quantity at a certain price. Excess lots, which occur as a result of the rounding down of the original allocated trade quantity, may be allocated on a first in, first out basis.

Production Environment
Production environment is the environment in which an automated trading system is operating including software and hardware used by traders, order routing to exchanges, market data, dependent databases, risk control systems, data capture and analysis systems, and post-trade processing systems.

Proximity Hosting
Proximity hosting is a third-party infrastructure hosting space that is located as close as possible to the execution facility's matching engine.

R

Regression Testing
Regression testing is a type of functional testing that confirms that no bugs are introduced into an existing system as a result of changes being made.

Repeated Automated Execution Limits
A repeated automated execution limit is the maximum number of times a strategy or identical order is filled and then re-enters the market without human intervention.

Responsible Party
A responsible person is a person who is designated either informally or by registering with an exchange or regulator as being the point of contact for information on the operation and control of an automated trading system.

Resting Order
A resting order is an order that has been submitted to the exchange but has not yet been executed. Resting orders are often placed using a limit price and are said to be passive since they do not trade immediately and will only trade when another participant aggresses to their price level.

S

Scalability Testing
Scalability testing is a type of non-functional testing that confirms the system performs as designed as it is extended to support use by more users or it is deployed to more environments.

SEC
Securities and Exchange Commission.

Security
Security consists of steps taken to limit the risk that software access becomes lost or controls disrupted, with potential market impact. In addition, steps are taken to protect employees and users from inadvertent compromise of the firm's data in order to avoid harm to the participant and the marketplace. Finally, steps are taken to provide for the appropriate access to development, testing, and production environments.

Self-Matched Trades
Self-matched trades are trades between the same or related entities that are matched in a central limit order book.

Self-Match Prevention Functionality
Self-Match Prevention (SMP) Functionality is a type of trading control designed to prevent a trader's order from inadvertently being matched against another of the participant's orders within the exchange's matching engine. SMP controls can be implemented differently due to the balance of flexibility versus complexity, and are often implemented differently across exchanges.

SMP Functionality
See Self-Match Prevention Functionality.

Software Development
Software development is the writing, testing, and maintaining of the source code associated with automated trading systems.

Software Development Environment
A software development environment is the combination of hardware, software, connectivity, and data required to develop the source code for an automated trading system.

Software Feature Requirements
Software feature requirements are a description of an automated trading system's required functionality provided to those responsible for the development and support of such systems.

Source Code
Source code is a collection of computer instructions written in a computer language used to create automated trading systems.

Source Code Implementation
Source code implementation is the process by which source code is written.

Source Code Review
Source code review is the process by which software engineers may have their source code reviewed for educational or verification purposes.

Source Code Testing
Source code testing refers to the work done by organizations to confirm that their trading systems and environments function as designed and within acceptable parameters.

Standards
Standards are a defined set of practices generally agreed to meet the target for quality in the creation, implementation, and operation of a system. The practices may be documented by a recognized standards body.

Standby Systems
Standby systems are hardware and software that is immediately available to be used for business continuity purposes should primary production systems encounter a problem. This may include computers, network circuits, and communication mechanisms.

Stop Logic Functionality
Stop logic is exchange-designed functionality that is designed to detect potential market movements caused by the triggering and trading of stop orders where the resulting price move would extend beyond an exchange specified threshold. When triggered, stop orders attempt to move the market to an executing price beyond a pre-established value, a Stop Logic event occurs. Stop Logic detects these situations and responds by placing the identified market in a state for a predetermined period of time where new orders may be accepted, but trades do not occur, providing an opportunity for market participants to respond to the demand for liquidity. See also Exchange Market Pauses.

Stop Price
A stop price is the price in a stop order that triggers creation of a market order. In the case of a sell-on-stop order, when the market price of the contract reaches or falls below the stop price a market sell order will be triggered for that contract. In the case if a buy-on-stop order, when the market price of the contract reaches or rises above the stop price, a market buy order will be triggered for that contract.

Stress Testing
Stress testing is a type of non-functional testing that confirms that the system operates in an acceptable manner during periods of atypical amounts of external inputs and internal events.

Supervisor
A supervisor is the head trader or other person responsible for an automated trading system

Surveillance System
A surveillance system is system that analyzes trade data to monitor for market participant behavior, as well as providing market performance information for an exchange.

Systems
Systems are the combination of software, hardware, and connectivity required to trade in an automated fashion.

T

Tabletop Review
A tabletop review is a discussion of possible risk scenarios that might occur and how the firm might react to them.

Tag 50
Tag 50 is a value within the FIX Protocol (known as "SenderSubID") that some exchanges may use for submission of operator IDs.

Tag 116
Tag 116 is an assigned value used within the FIX Protocol (known as "OnBehalfOfSubID") to identify specific message originator (i.e., trader) if the message was delivered by a third party, for example, a third-party vendor or FCM.

Testing
See Source Code Testing or individual types of testing.

Trade Cancellation
Trade Cancellation is a cancellation of an executed trade made in accordance with an exchange's error trade rule or policy.

Trader ID
A Trader ID is an identifier attached to an order that uniquely identifies the participant submitting the order, often submitted in Tag 50 or Tag 116 of the FIX Protocol.

U

Unit Testing
Unit testing is a type of testing in which discrete units of source code are tested to verify they work as desired. These tests may be configured to run automatically throughout the development process.

V

Velocity Logic
Velocity logic is designed to detect market movement of a predefined number of points either up or down within a predefined time. Velocity logic responds by placing the identified market in a state for a pre-determined period of time where new orders may be accepted, but trades do not occur, providing an opportunity for market participants to respond to the demand for liquidity. See Exchange Market Pauses.

2001 Pennsylvania Avenue N.W., Suite 600
Washington D.C. 20006-1823
Phone: 202.466.5460 | Fax: 202.296.3184
FIA.org