



DMIST

DERIVATIVES MARKET
INSTITUTE FOR STANDARDS

Execution Source Code Standard

August 19, 2026

TABLE OF CONTENTS

DISCLAIMER.....	2
1. ACKNOWLEDGEMENTS	3
2. EXECUTIVE SUMMARY	4
3. PURPOSE OF THE STANDARD	6
4. SCOPE AND APPLICABILITY	7
5. BENEFITS.....	7
6. HIGH LEVEL FLOW	8
7. FINAL STANDARD	9
GENERAL RESPONSIBILITIES	9
CLIENTS/END USER RESPONSIBILITIES	10
INDEPENDENT SOFTWARE VENDORS (ISV) RESPONSIBILITIES.....	10
EXECUTING/CARRY BROKER RESPONSIBILITIES	11
EXCHANGE / EXCHANGE GATEWAY RESPONSIBILITIES.....	12
CLEARING HOUSE / CCP RESPONSIBILITIES.....	12
CLEARING BROKER RESPONSIBILITIES	13
PERMISSIBLE VALUES.....	14
EXCLUSIONS / KNOWN LIMITATIONS.....	15
8. IMPLEMENTATION AND ADOPTION	16
9. METRICS	17
10. GLOSSARY.....	18
11. QUESTIONS/FURTHER INFORMATION.....	20
APPENDIX.....	21

DISCLAIMER

This Final Standard is intended for informational and educational purposes only and is not intended to provide investment, tax, business, legal or professional advice. FIA and DMIST recommend seeking independent expert advice where needed. FIA, DMIST, nor its or their members/participants endorses, approves, recommends, or certifies any of the information, opinions, products, or services referenced in this Final Standard. FIA and DMIST make no representations, warranties, or guarantees as to this Final Standard's contents. Neither FIA nor DMIST accept any responsibility for anyone placing reliance upon this Final Standard.

1. ACKNOWLEDGEMENTS

DMIST was formed as an outgrowth of industry conversations following high volume and volatility in February and March 2020, at the onset of the COVID-19 pandemic. DMIST's ultimate goal is to encourage widespread adoption of standards in the Exchange-traded derivatives industry that will help make markets more efficient, resilient, and competitive for all.

There are two levels of participation in DMIST: (1) Sponsor Board members who consider and approve standards; and (2) Ambassador members, including technology vendors, who are subject matter experts and who help identify, develop, and calibrate standards for the Sponsor Board's approval. For more information on participating in DMIST, [click here](#). More information regarding the history and development of DMIST is available at: [Derivatives Market Institute for Standards \(DMIST\)](#).

2. EXECUTIVE SUMMARY

The Derivatives Market Institute for Standards (DMIST) was established to encourage the widespread adoption of standards that improve efficiency, resilience, and competitiveness across the exchange-traded derivatives ecosystem. A foundational component of this mission is the consistent identification and persistence of critical trade reference data throughout the full trade lifecycle. The Execution Source Code Standard addresses a long-standing industry challenge: the lack of a reliable, end-to-end method to identify how a trade was executed and ensure that information is preserved from execution through clearing and downstream processing.

Accurate identification of execution source is essential to the correct application of brokerage rates, efficient reconciliation, and high-quality downstream processing. While industry participants have broadly recognized FIX Protocol Tag 1031 as a mechanism for identifying execution source – and while the FIA Execution Source Code Schema has existed since 2019 – adoption across exchanges, CCPs, and market participants has been inconsistent and incomplete. In many cases, execution source values are missing, overwritten, or unvalidated, rendering them unreliable for systematic use and forcing firms to rely on manual processes, bilateral mappings, and exception handling.

These deficiencies create operational friction that becomes increasingly problematic as the industry advances toward greater automation, interoperability, and more time-sensitive workflows. As billing, allocations, risk, and clearing processes accelerate toward near-real-time models, tolerance for ambiguous or late-arriving execution source data continues to diminish. Industry feedback gathered through DMIST's consultation process underscored that execution source must be treated not as optional metadata, but as core reference data that underpins accurate billing, reconciliation, and scalable straight-through processing.

In response, this Final Standard establishes clear expectations for the identification, validation, and end-to-end persistence of execution source across both FIX and non-FIX workflows. The standard clarifies the responsibilities of each participant type including clients, executing brokers, exchanges, CCPs, clearing brokers, and vendors. It also emphasizes consistent population at the point of execution and uninterrupted transmission throughout the listed derivatives lifecycle. By addressing accountability, data integrity, and cross-platform applicability, the standard aims to move execution source from inconsistent practice to dependable, industry-wide infrastructure.

Broad adoption of this standard is expected to materially reduce operational risk, minimize brokerage errors, and decrease manual reconciliation and exception handling. More importantly, it provides a scalable data foundation aligned with DMIST's broader framework and the industry's long-term evolution toward real-time processing and improved interoperability. The Execution Source Code Standard is a critical step toward eliminating legacy inefficiencies and enabling a more transparent, automated, and resilient derivatives ecosystem.

3. PURPOSE OF THE STANDARD

The purpose of the DMIST Execution Source Code Standard is to establish a clear and consistent framework for capturing, validating, transmitting, and preserving execution source as authoritative reference data across the listed derivatives lifecycle. The standard is intended to ensure that a trade's execution source is populated at the point of execution, retained without reinterpretation or overwrite, and made available for downstream operational, billing, reconciliation, and reporting processes.

Accordingly, the standard defines baseline expectations for all relevant participants across both FIX and non-FIX workflows, including responsibilities for data integrity and end-to-end persistence. Its aim is to support accurate brokerage treatment, reduce manual intervention and avoidable exceptions, and provide a durable foundation for greater automation and interoperability across the exchange-traded derivatives ecosystem.

4. SCOPE AND APPLICABILITY

This standard applies to all exchange-traded derivatives transactions, regardless of asset class, execution method, protocol, or geographic region. It is intended to be implemented consistently by all participants involved in the trade lifecycle, including clients/end users, executing and carry brokers, exchanges, central counterparties (CCPs), clearing brokers, and technology vendors.

The standard is technology-agnostic and applies equally to FIX, API-based, and proprietary connectivity models. Its requirements are designed to support both current-state workflows and the industry's continued evolution toward increased automation and real-time processing.

5. BENEFITS

Key benefits of a standardized approach to Execution Source Code are:

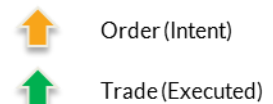
- Accurate identification of execution method;
- Reduced manual processing and trade amendments;
- Improved reconciliation and exception management;
- End-to-end consistency across the trade lifecycle;
- Technology-agnostic and scalable implementation;
- Correct and timely brokerage and fee calculations;
- Faster and more reliable settlement of brokerage; and
- Foundation for greater automation and future-state workflows.

6. HIGH LEVEL FLOW

The DMIST Execution Source Code Working Group outlined the end-to-end workflow to illustrate each market participant's role in the generation and transmission of the Execution Source Code.



Figure 1 - Workflow by Participant Type



7. FINAL STANDARD

Execution Source Code is a single identifier that captures the Client/End User's intended execution method. Execution Source Code may be transmitted via industry-standard protocols such as FIX, or via API or binary connections. Where the FIX Protocol is used, Tag 1031 is the appropriate field. Where API or binary connections are used, participants should use the values set forth in this standard. The success of this standard relies on the ongoing support of six (6) key participant types, each of which plays a critical role in improving brokerage and fee processing.

GENERAL RESPONSIBILITIES

- 7.1 Each participant should persist the Execution Source Code as authoritative reference data. Passing the field without retaining it for downstream processing, audit, reconciliation, or billing purposes does not meet the intent of this standard.
- 7.2 Execution Source Code should persist unchanged from the point of execution through all downstream processing, including, but not limited to:
 - Allocations
 - Average pricing
 - Give-ups and take-ups
 - Trade splits
 - Clearing, settlement, and reporting events
- 7.3 Processing events should not overwrite, default, reinterpret, or deduce Execution Source Code values unless explicitly instructed by the originating participant (Client) with appropriate controls and auditability.

CLIENTS/END USER RESPONSIBILITIES

Clients/end users are the driving force behind the Execution Source Code standard. It is their brokerage and fees who are processed using this code.

7.4	Clients should provide intended method of execution on orders to any system or party who will be involved in executing the order.
7.5	Clients should utilize Execution Source Codes for which they have a valid rate schedule agreed.
7.6	Clients should configure O/EMS and any other system to capture the proper execution method.
7.7	Clients should transmit the Execution Source Code received back from the Exchange to Clearing Brokers on allocations.
7.8	Clients should align internal processes to the Execution Source Code values outlined in the table of <u>Permissible Values</u> .
7.9	Clients should coordinate with Executing Brokers/Carry Brokers to ensure agreement on use of values in the table of <u>Permissible Values</u> .

INDEPENDENT SOFTWARE VENDORS (ISV) RESPONSIBILITIES

The role of ISVs is key in technically supporting the ESC fields in their platforms. This will allow downstream participants to rely on a consistent identifier for execution method and fee treatment. It is crucial that ISVs pass the code consistently to the EB, exchanges, and downstream platforms.

7.10	ISVs should support the consumption and transmission of the Execution Source Code.
7.11	ISVs should support Execution Source Code in data schemas, both upstream and downstream.
7.12	ISVs should not override Execution Source Code values with any value that changes the original intent of the Client.
7.13	ISVs should not conduct any sort of proprietary remapping in the original Execution Source Code field.
7.14	ISVs should implement logging and audit trails for Execution Source Code sufficient to allow participants to audit data usage and propagation.
7.15	ISVs should support Execution Source Code as a standard data element across order entry, execution, allocation, clearing, or post-trade processing, as applicable.

7.16	ISVs should enable population, persistence, and transmission of Execution Source Code in alignment with this standard.
7.17	ISVs should avoid deducing or altering values without explicit instruction from the originating participant (Client).

EXECUTING/CARRY BROKER RESPONSIBILITIES

The role of the Executing Broker is crucial in the adoption and advocacy of the Execution Source Code standard through improvements in brokerage and fee processing. They are key in aligning the Client's intent and the actual execution method at the Exchange.

7.18	Executing Brokers should support the consumption and transmission of the Execution Source Code on all orders.
7.19	Executing Brokers should not override Execution Source Code values sent from the Client or upstream ISV(s) with any value that changes the original intent of the Client.
7.20	Executing Brokers should implement logging and audit trails for Execution Source Code sufficient to allow participants to audit data usage and propagation (e.g., blank or inconsistent values).
7.21	Executing Brokers should avoid deducing, defaulting or altering values without explicit instruction from the originating participant (Client).
7.22	Executing Brokers should provide sufficient logging and transparency to allow participants to audit data usage and propagation.
7.23	Executing Brokers should not conduct any remapping that overrides the Client and should transmit the Client's intended execution method. For example, if a Client wishes to execute via an execution desk where the Executing Broker is not a member of the selected Exchange, the Execution Source Code must be W even if the execution method between the Executing Broker and Member (Carry) Broker differs.
7.24	Carry Brokers, when used, should not override the Execution Source Code intended by the originating participant (Client).

EXCHANGE / EXCHANGE GATEWAY RESPONSIBILITIES

Exchanges play a critical role in broadening the adoption of the Execution Source Code standard. Once a trade is executed, the Exchange becomes the authoritative distributor of the method of execution.

7.25	Exchanges should support Execution Source Code in data schemas, both upstream and downstream.
7.26	Exchanges should require population of Execution Source code on all trades.
7.27	Exchanges should transmit the Execution Source Code to the relevant upstream consumers (Executing Broker, ISV, Client).
7.28	Exchanges should transmit Execution Source Code to relevant Clearinghouse / CCP via data feeds or other transmission methods.
7.29	Exchanges should consume the Execution Source Code on all orders.

CLEARING HOUSE / CCP RESPONSIBILITIES

Central counterparties play an essential role in preserving Execution Source Code as trades transition from execution to clearing.

7.30	CCPs should support Execution Source Code in data schemas, both upstream and downstream.
7.31	CCPs should consume Execution Source Code from Exchanges without modification or inference.
7.32	CCPs should persist Execution Source Code through novation, netting, and clearing processes.
7.33	CCPs should transmit Execution Source Code to Clearing Brokers on all relevant clearing and trade reporting feeds.
7.34	CCPs should ensure that lifecycle events, including give-ups, allocations, and trade splits, do not result in the overwriting or loss of Execution Source Code.
7.35	CCPs should escalate systematic data quality issues to upstream participants where execution source is consistently missing or invalid.
7.36	CCPs should not allow modifications to the Execution Source Code after execution that change the original intent of the Client.

CLEARING BROKER RESPONSIBILITIES

Clearing Brokers play a critical role in the application and settlement of brokerage and fees on behalf of Clients and Executing Brokers.

7.37	Clearing Brokers should consume Execution Source Code from clearinghouses / CCPs without modification or inference.
7.38	Clearing Brokers should persist Execution Source Code through all downstream processing.
7.39	Clearing Brokers should consume the Execution Source Code as the definitive determinant for applying brokerage rates. Proprietary logic such as combining trader IDs, platforms or inference should be avoided.
7.40	Clearing Brokers should utilize the Execution Source Code to accurately bill Clients, accrue and pay Executing Brokers, and manage brokerage billing on behalf of third parties.
7.41	Clearing Brokers should highlight and escalate persistent upstream data quality issues that materially impact billing accuracy or reconciliation outcomes.

PERMISSIBLE VALUES

DMIST recognizes that execution models have and will continue to evolve over time, necessitating an expansion of permissible values. As such, the following principles apply to permissible values:

- 7.42 The current permissible value set should be supported by all participants in the workflow.
- 7.43 A single character, multiple string value, letters or numbers, should be used to represent the execution source method (A-Z, 0-9).

A comprehensive table of permissible values will be set forth in the Execution Source Code Implementation Guide and will be governed by DMIST via collaboration across the industry to avoid discretionary interpretation and fragmentation.

The following table outlines the current permissible values:

Code	Execution method definition	When to use
W	Desk (high-touch)	Use W for broker-desk / high-touch execution.
H	Premium algorithmic trading provider	Use for Client self-executed flow via a premium algo provider.
C	Vendor-provided / third-party platform	Use for Client self-executed flow via a third-party software or vendor platform.
G	Sponsored access	Use for Client self-executed flow via sponsored access.
Y	Other Client self-executed electronic flow	Use for all other low-touch electronic flow, such as flow using the Executing Broker's infrastructure.
D	Other	Optional catch-all for "other" electronic scenarios.

Table 1 - Table of Permissible Values

EXCLUSIONS / KNOWN LIMITATIONS

1. Execution Source Code (including FIX Tag 1031) is intentionally defined as a simple, single identifier to promote adoption and consistency. Market participants acknowledge that a single field cannot:
 - a. Represent multiple execution legs simultaneously.
 - b. Fully capture party roles, platform usage, or billing hierarchies in complex scenarios.
2. Accurate reflection of rate types on Give-Up Agreements rate schedules is critical.
3. Any expansion of the permissible values should be managed with clear governance to avoid discretionary interpretation and fragmentation.

8. IMPLEMENTATION AND ADOPTION

The effectiveness of this standard is materially dependent on consistent and accurate population of Execution Source Code across all participating venues and intermediaries. Industry experience demonstrates that ad hoc adoption alone is insufficient to achieve reliable data quality at scale. DMIST encourages a phased adoption model, whereby:

- Execution Source Code population may initially be optional to allow firms to implement required system changes; and
- Population subsequently becomes mandatory once a reasonable implementation window has elapsed.

Where Execution Source Code is mandatory, market operators and intermediaries should implement clear enforcement mechanisms, which may include:

- Message validation controls;
- Soft warnings during transition periods; and
- Escalation toward rejection or remediation requirements for persistently inaccurate or missing values.

Inaccurate, missing, or inconsistent Execution Source Code creates commercial, operational, and financial risk, particularly in brokerage billing. As such, accountability for data quality must be clearly assigned to the participant responsible for population at each stage of the lifecycle.

Governance and Evolution of Value Sets

A comprehensive table of permissible values will be set forth in the Execution Source Code Implementation Guide and will be governed by DMIST via collaboration across the industry to avoid discretionary interpretation and fragmentation.

9. METRICS

Metrics should be generated and consistently reviewed to allow the industry to continually monitor progress against the Execution Source Code Standard, identify shortcomings and recommend strategic improvements.

Implementation and adoption metrics include:

1. Percentage of trades populated with a valid Execution Source Code by venue;
2. Percentage of trades where Execution Source Code is overridden downstream;
3. Coverage by trade type, including give-ups and block trades;
4. Percentage of brokerage adjustments attributable to execution method discrepancies.

Metrics may be sourced from Exchanges, CCPs, Clearing Brokers, or industry utilities and should be captured quarterly and reviewed periodically to:

- Assess adoption progress;
- Identify systemic data quality issues; and
- Support the business case for broader mandates.

10. GLOSSARY

“Algorithmic Trading” – A type of transaction that uses computer programs to buy or sell.

“Broker Intermediated” – A type of transaction whereby trading uses a go-between who connects a Client with an Exchange rather than a Client interacting directly with an Exchange.

“CCP” – A central counterparty is a financial institution that interposes itself between counterparties to trades that have been executed at an Exchange. It becomes the buyer to every seller and the seller to every buyer, takes on the counterparty risk and provides clearing and settlement services to its customers/members.

“Clearing Broker” – A financial firm or operator that serves as an intermediary between a Client’s Executing Broker and the central counterparty.

“Clearing Member” – A firm meeting the requirements of, and approved for, clearing membership at the Exchange.

“Client” – An individual or organization, typically an end-user, asset manager, proprietary trading firm or similar party, who initiates an order to buy or sell a product in the Exchange-traded market.

“Client Initiated” – A type of transaction where an order originates directly from the Client rather than being suggested by a broker or advisor.

“Exchange” – A financial exchange where participants can trade (buy/sell) standardized products as defined and listed at the exchange. An exchange will have a relationship with a CCP to facilitate the clearing and settlement of the trades.

“Executing Broker” – A financial firm or operator responsible for finding market liquidity and buying and selling on behalf of a Client.

“Execution Source Code” – Execution Source refers to the Client’s intended method of execution for a given trade, captured as a single standardized identifier and treated as authoritative lifecycle reference data.

“FIA Execution Source Code Schema” – The best practice introduced in 2019 by FIA and FIA Tech that outlines how to use FIX Tag 1031 to identify execution source method.

“FIX Protocol” – Financial Information Exchange, a global standard messaging protocol for electronic trading.

“High Touch” - High-touch execution refers to trades executed with material broker intervention, including sales-trader or desk-based execution.

“ISV” – Independent Software Vendor. A company that develops and sells software applications that provide a specific service in the Exchange-traded derivatives lifecycle.

“Low Touch” - Low-touch execution refers to Client-initiated electronic execution, including algorithmic trading, third-party platforms, sponsored access, or broker-provided electronic infrastructure.

“Participant” – Any party in the lifecycle of Exchange-traded derivatives.

“Tag 1031” – In FIX Protocol, the field used to identify the execution source method. Also known as CustOrdHldInst.

“Web API” – An application processing interface between a web server and a web browser. Also known as Web Service API.

11. QUESTIONS/FURTHER INFORMATION

If you have questions about this standard, wish to know more about DMIST or have an interest in joining the initiative, please visit the [DMIST website](#). You may also contact Don Byron at dbyron@fia.org or Staci Parrish sparrish@fia.org.

APPENDIX

DMIST Overview

DMIST was formed as an outgrowth of industry conversations following high volume and volatility in February and March 2020, at the onset of the COVID-19 pandemic. DMIST's ultimate goal is to encourage widespread adoption of standards in the Exchange-traded derivatives industry that will help make markets more efficient, resilient, and competitive for all.

There are two levels of participation in DMIST: (1) the Sponsor Board members who consider and approve standards; and (2) the Ambassador level members, including technology vendors, who are subject matter experts and who help identify, develop and calibrate standards for the Sponsor Board's approval. For more information on participating in DMIST, please visit our website.

For more information regarding the history and development of DMIST, please see Modernizing the Listed Derivatives Workflow: A Blueprint for Change (November 2021) and DMIST's 2023 Annual Progress Report.

DMIST Standard Process

The process that DMIST follows for a proposal to become a standard is:

Step One: Proposed standard is received from any member of the public (including, but not limited to, Sponsor Board Members or Ambassadors).

Step Two: Sponsor Board determines whether the initial proposal meets certain required criteria (e.g., the submission contains sufficient requisite information, the proposed standard addresses a topic that is considered in scope for DMIST's consideration, the proposed standard relates to a topic that would significantly benefit the industry to standardize).

Step Three: DMIST forms Ambassador Working Group(s) for review and consideration of the proposed standard.

Step Four: Sponsor Board meets to consider the feedback of the Ambassador Working Group(s) and votes to publish proposed standard for public comment.

Step Five: DMIST issues the proposed standard for public comment.

Step Six: At the end of the comment period, the Ambassador Working Group(s) used to vet the standard as initially proposed will meet to review and consider the comments.

Step Seven: Sponsor Board votes to approve the standard based upon feedback and recommendations from the Ambassador Working Group(s). If approved, the standard is considered final and is published.

Step Eight: The success of DMIST requires commitment to transparency with regards to adoption and implementation of standards. Each final, approved standard will specify transparency expectations for Sponsor Board Members regarding whether or not they have chosen to adopt or implement such a standard. Each final approved standard must also specify metrics, to be measured on an ongoing basis, related to implementation, adoption, usage, and/or effectiveness.

DMIST

DERIVATIVES MARKET
INSTITUTE FOR STANDARDS

2001 K Street, NW
Suite 725, North Tower
Washington, DC 20006

Visit our [website](#)