SECURITIES AND FUTURES COMMISSION
證券及期貨事務監察委員會

Data Standards for Order Life Cycles (“DS-OL”)

July 2019

Contents

Terms/Abbreviations

Terms/Abbreviations	Description
Algo	Algorithmic Trading System or Engine
ALP	Alternative Liquidity Pool
ASCII	American Standard Code for Information Interchange
CBBC	Callable Bull / Bear Contracts
CD	Careful Discretions
CUSIP	Committee on Uniform Security Identification Procedures
DS-OL	Data Standards for Order Life Cycles
Execution Venue	This can be a market-side execution venue (i.e. an exchange or another broker) or an internal execution venue (e.g. internal cross, ALP match and manual fill).
FIX	Financial Information eXchange, which is a well-recognized and common standard used in equity electronic trading globally
GTC	Good-til-cancelled (order time validity)
GTD	Good-til-date (order time validity)
High Touch	This refers to trade flows that require human intervention from order inception to processing, and eventually order amends, cancels and/or executions
HKEX	Hong Kong Exchanges and Clearing Limited
In-Scope Broker	Broker required to implement the DS-OL (please refer to Paragraph 6 for details)
IOI	Indication of Interest
Internalised Trade	This refers to a trade that is effected on an internal execution venue or involves allocation of executions. (Please refer to Section 3.1.7.1 for details)
ISO	International Organization for Standardization
ISIN	International Securities Identification Number
LEI	Legal Entity Identifier
LOID	Logical Order ID
Low Touch	This refers to electronic trade flows that require little to no human intervention from order inception to processing, and eventually order amends, cancels and/or executions
Material Terms	Minimum information about a client order that In-Scope Brokers are required to store (please refer to Section 2.3 for details)
MIC	Market Identifier Code
OMS	Order Management System
QUIK	Quick Code being the security code assigned in Japan for listed companies
Reference Data Dictionary	A dictionary of the In-Scope Broker which maps the names of accounts, order types or instructions, etc. to their full names and/or detailed description
RIC	Reuters instrument code
SEDOL	Stock Exchange Daily Official List
SEHK	The Stock Exchange of Hong Kong Limited

SFC	Securities and Futures Commission
UTC	Coordinated Universal Time
VWAP	Volume Weighted Average Price

1. Introduction

1.1 Background

1. During our recent inspections of active securities brokers, the performance of data analytics (where multiple tests could be run automatically to identify irregular transactions) had proved to be very effective and led us to detect various systemic trading-related control deficiencies and instances of non-compliance with the Code of Conduct[1].

2. However, due to the lack of an industry-wide convention, data collection was inefficient and required substantial time and efforts from both the brokers as well as Commission staff. In particular, none of the brokers' existing systems could readily reconstruct order life cycles from initial receipt of an order instruction[2] to final execution or cancellation. This was not satisfactory as such information was particularly useful for identifying irregular transactions.

3. For the past 16 months, the SFC has been working in close collaboration with an external consultant and a working group of selected licensed brokers in developing data standards aimed to facilitate data analytics performed by the SFC. This involves developing an overall framework, determining the scope of application and setting detailed technical specifications governing how order and trade data (including prescribed minimum content) should be presented to the SFC in a meaningful and consistent manner.

1.2 Overall framework and scope of the present data standards

4. It was agreed that a phased approach should be adopted, starting with a fairly limited scope, whilst the data standards should be capable of potential expansion in future phases to enable additional compliance testing covering other products and trading activities.

5. On the premise that trading misconduct has historically been associated with transactions involving equity more than other types of securities or futures contracts, the data standards initially would only cover equities listed on the SEHK ("In-Scope Products").

6. Only brokers with top trading volumes in the In-Scope Products should comply with the data standards as data analytics should be more relevant to them than to other brokers. For this purpose, an activity-level based methodology has been developed based on a broker's trading turnover in SEHK-listed equities[3] in any calendar year (in $ terms) ("Annual Trading Turnover") as compared to that year's total market trading volume (i.e. as published annually in the HKEX Market Statistics Report[4]). Where a broker's

[1] The Code of Conduct for Persons Licensed by or Registered with the Securities and Futures Commission ("Code of Conduct")

[2] The terms "order instruction", "order" and "client order" are used interchangeably in this document. Unless stated otherwise, an order may be received from a client or generated internally by a broker and the word "client" should be construed accordingly.

[3] This covers trades in listed equities that are either executed in the SEHK or get reported into the SEHK (for example, transactions executed in ALP or by way of internal crosses of client orders)

[4] Please refer to an example at https://www.hkex.com.hk/-/media/HKEX-Market/Market-Data/Statistics/Consolidated-Reports/Annual-Market-Statistics/2018-market-statisitcs.pdf

Annual Trading Turnover reaches or exceeds 2%[5] of the total market trading volume for that same year (hereafter referred to as "the 2% threshold"), the broker is classified as an In-Scope Broker. Once classified as an In-Scope Broker, it will have to comply with the data standards thereafter on an ongoing basis, even if its Annual Trading Turnover has subsequently fallen below the 2% threshold.

7. One of the key objectives of the data standards would be to facilitate the reconstruction of order life cycles, from the time a client order is received through to its final execution and/or cancellation. As such, the Data Standards for Order Life Cycles ("DS-OL") were developed to decompose the order life cycles into discrete events and assign a unique logical order identifier to each top-level parent order[6], i.e. logicalOrderID ("LOID"), so as to provide transparent linkage to all subsequent events.

8. When designing the DS-OL, special care was also taken to provide maximum flexibility when causing minimum disruption, especially for In-Scope Brokers forming part of a global conglomerate subject to group policies and procedures. As such, the DS-OL do not prescribe any specific implementation requirements, but only define the content and format of data which In-Scope Brokers should be able to readily present. In-Scope Brokers are therefore not required to modify their existing systems in order to comply with the DS-OL. In addition, local and international regulations and standards were taken into account and the DS-OL make reference to FIX to leverage on some fields commonly used in FIX when defining certain terms in the DS-OL.

9. Separately, data submission instructions and general guidelines to assist In-Scope Brokers in validating the contents of data in the DS-OL format are provided in the Appendix.

[5] For additional information, we have considered setting the mark at different percentages based on 2016-2018 HKEX transaction data. 2% is considered most appropriate as that captures all Category A and a few Category B SEHK Exchange Participants, and together they accounted for more than 55% of the total market trading volume of those three years respectively.

[6] This is interchangeable with "parent order", "top parent order" in this document, unless otherwise stated

2. Key Features

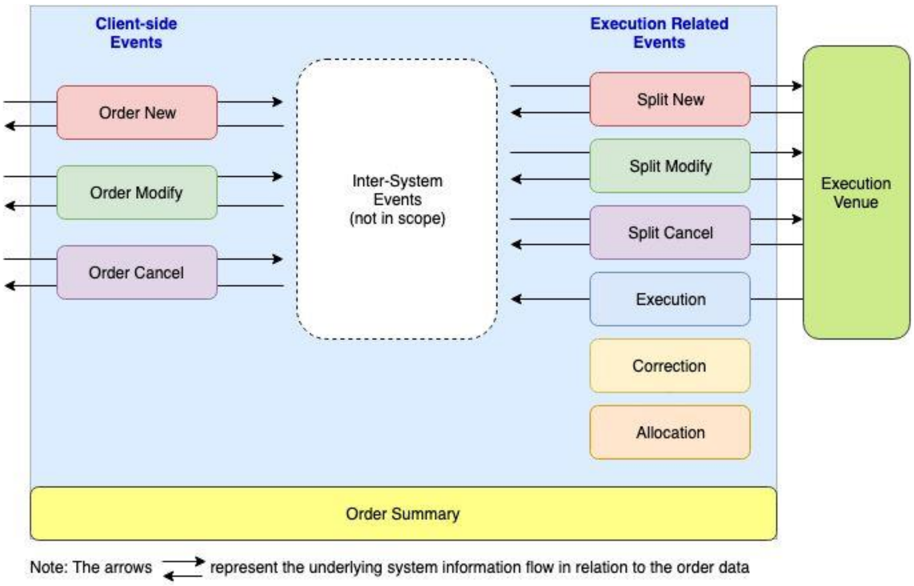
2.1 Summary of the DS-OL

10. The DS-OL are only concerned with an order life cycle from the time a client order is received through to processing and eventually execution and/or cancellation. In other words, the DS-OL only cover pre-trade and at-trade information; post-trade information such as allocations to the underlying funds or sub-accounts under the same client, settlement and fees and commissions are not in-scope.

11. The DS-OL also adopt an event-based approach to reconstruct the order life cycles. Events which are to be recorded in the DS-OL format are mainly the client-side events (e.g. client requests[7] to place a new order, modify or cancel an order) and execution-related events (e.g. child order splits to Execution Venues and the executed trades).

12. It should be noted that these events are composite representations of an order's progression. Thus, it is possible that creating an event may require information from multiple systems or databases, and there is no 1-1 relationship between system messages and the DS-OL event records. Events are therefore not analogous with system records e.g. audit logs or system messages. Also, in most cases, internal events such as the passing of an instruction from one intermediate system to another (e.g. order passing from an OMS to an Algo), especially in instances where there is no modification and it is simply a 'pass-through' of data) are not in-scope. Hence, when a typical trading infrastructure may involve multiple trading systems to process an order, the DS-OL do not seek to capture what happens in-between or the so-called 'intermediate' order statuses or replicate system messages.

Figure 1.1: The Order Life Cycle



[7] The terms "request" and "instruct" are used interchangeably in this document.

2.2 Trading Systems Data Requirements

13. Reconstruction of an order life cycle may involve extracting information stored in single or multiple trading systems or components. The DS-OL do not make requirements as to the specific systems which should be the source of the DS-OL information for an In-Scope Broker (e.g. OMS, middle- or back-office systems, and/or other systems). This ambiguity is intentional. Given that each In-Scope Broker has unique implementations and technical architectures supporting its business activities, any and all systems operated by an In-Scope Broker may be considered as in-scope and valid sources of information, provided that data extracted from these systems can comply with the DS-OL.

14. An In-Scope Broker can choose to generate the required data in the DS-OL format from its own format only upon the SFC's request or already store them separately in the DS-OL format. The implementation of tools to ensure that data can be provided to the SFC in the DS-OL format on a timely basis is left to the discretion of each individual In-Scope Broker. It is important, however, to note that although some internal system events and data like algorithm parameters stipulated by clients do not need to be submitted in the DS-OL format, In-Scope Brokers should still keep records of these events and information in order to be able to fully explain an order life cycle.

2.3 Material Terms of an Order

15. The DS-OL require In-Scope Brokers to record Material Terms (i.e. minimum information) of an order and/or any changes to these terms throughout an order life cycle. Material Terms of an order shall include, but are not limited to:

- Order price
- Order quantity
- Order side
- Order type and instructions
- Time in force (or time validity)
- Order capacity
- Client identification
- Algorithmic order instructions (i.e. algorithm parameters according to the In-Scope Broker's system settings for algorithmic orders[8])
- Aggregating information behind aggregated orders

16. In general, if any Material Terms of an order is modified, a corresponding modification event (Order Modify Event) should be recorded in the DS-OL format.

2.4 Timestamps

17. Timestamps are required to differentiate events and allow for accurate reconstruction of the sequence of events. Timestamps, in the DS-OL, refer to timestamps provided by the In-Scope Broker from its internal systems (not by

[8] The DS-OL do not seek to assess the inner mechanics of an In-Scope Broker's proprietary algorithms or Algo.

an exchange). The DS-OL accept timestamps with up to 100 nanosecond. In-Scope Brokers should provide timestamps with granularity appropriate for the type of trading activity being reported. To the extent that timestamps are not granular enough to differentiate between individual events or accurately reconstruct the sequence of those events, a time sequence can be provided using the eventSequence field.

2.5 Order Linkage

18. In-Scope Brokers are required to effectively link order instructions with corresponding events. Depending on the implementation of an In-Scope Broker's systems, an order which is received from a client or routed to an Execution Venue; may have no, one or multiple identifiers within its systems' messaging protocol. Additionally, identifiers may be mutable throughout an order life cycle, or may be reused over time and applied to subsequent orders.

19. To facilitate event linkage, the DS-OL require the use of a logical order identifier, the LOID which identifies the top-level parent order. This identifier is immutable and recorded alongside all the events throughout that order life cycle, allowing transparent linkage between all these events including child /grandchild order splitting, and modifications / cancellations / executions for the parent order and any of its sub-generations.

20. Where a client order is first received by another licensed securities broker that is a group company of an In-Scope Broker and this broker relays the order to the In-Scope Broker, this order should be assigned the same LOID. However, this will not be necessary where the client order is first received by a broker that is not a group company or is a registered institution.

3. Technical Specifications: Events

3.1 Decomposing to Order Life Cycle Events

21. To reconstruct and report order life cycles under the DS-OL, an order life cycle is decomposed into various event types. They are: Order New, Order Modify, Order Cancel, Split New, Split Modify, Split Cancel, Execution, Execution Correction and Allocation. Separately there is also the Order Summary event which represents a summation of the life cycle after completion. Each of these events contains several data fields.

22. A full list of data fields and the corresponding details are listed in Section 3.3 - “Field Definitions” when the field requirement by each event type could be found in Section 3.4 - “Field Requirement Status by Event Type”.

3.1.1 Order New (ONEW) Event

23. The Order New event denotes the ’starting point’ of each order (for a single security name) received by the In-Scope Broker from a client. For an order which may exist within multiple trading systems or technology stacks within an In-Scope Broker, only one ONEW event is required. Hence for different business flows within the In-Scope Broker, this starting point can vary, and the In-Scope Broker needs to identify these starting points accordingly based on its own business flows.

24. Order New is a consolidated event capturing data from any of (1) the new order message or action, (2) acknowledgement or reject message, (3) order ticket record, and (4) any other client or market static data, if applicable.

25. The time of order receipt means the time the In-Scope Broker assigns an orderID to an incoming message. Each Order New event needs to have a LOID which will be used as the linkage to all subsequent events of this order. LOID is required to be unique and immutable across the entire dataset submitted by an In-Scope Broker to the SFC. A possible method of generating a LOID is a ’Date prefix’ + ’orderID’, i.e. YYYYMMDD_<OrderID>. Although the orderID should be unique during a trading day, orderID may be reused over time.

3.1.1.1 Order Type and Order Instructions

26. Orders received by an In-Scope Broker may be one of several different ’types’ of orders specifying different behaviours, such as price limitations, conditional logic and targeting a benchmark price, etc. The types of orders offered by In-Scope Brokers differ widely across organisations. To capture such instructions in a consistent manner, the orderType and orderInst fields are used where orderType categorises the order into one of several types as defined in the orderType enumerations and orderInst is a dictionary look-up value which maps the name of an order type or instruction (e.g., CD order) to its full name and/or detailed description as per the In-Scope Broker’s Reference Data Dictionary.

27. Multiple instructions should be separated using '|'[9] as a delimiter. E.g. 1) a 'special limit order' could have orderType=LMT and orderInst=special-limit; 2) a 'CD order, stay on bid-side only' could have orderType=MKT and orderInst=CD|stayonbid. The instruction string keys can be freely defined and should map to the In-Scope Broker's Reference Data Dictionary. A filler value is required when not applicable.

3.1.1.2 Multi-Day Orders

28. Orders received by In-Scope Brokers that remain in force for more than one day such as GTC, GTD, etc. are referred to as 'multi-day' orders. When presenting multi-day orders, the LOID is required to remain unchanged during their life cycles, regardless of how multi-day orders are implemented in the In-Scope Broker's systems (e.g., by automatically recreating a new order every business day or rolling over the same order to the next business day).

29. The treatment of order quantities (orderQty) may differ across systems with respect to multi-day orders. To support systems which do not modify the original orderQty during the order life cycle, the nDayOrderQty field should be used to indicate the outstanding orderQty.

3.1.1.3 Trading Capacity of Agency and Principal Orders

30. The 'Order Capacity' (orderCapacity)[10] of a new order should indicate the intended trading or executing capacity of the order. E.g., if a client requests the In-Scope Broker to provide a guaranteed price (a guaranteed VWAP order for example), the orderCapacity should be 'Principal' when it is recorded in the Order New event.

31. If the orderCapacity changes (e.g. from agency to principal) after an order has been received, this should be recorded with an Order Modify event. In this case, individual Order and/or Split events will be denoted as agency or principal as the case may be, and in the Order Summary, the orderCapacity will be mixed (value 'AP' which is only applicable in an Order Summary event).

3.1.1.4 Basket Orders, Multi-Leg Orders and Strategies

32. For any order which is part of a collection of orders, (e.g. a basket order, or a multi-leg order/strategy) there is an additional field, collectionID, in the Order New event which is used to indicate such a relationship with other orders in the same collection.[11]

33. The collectionID is composed of two parts using '='[12] as a separator. The first part is a free text field with an indicative name for the type of a collection when the second part is an identifier for individual collections which must be unique on each trading day. Examples of the collectionID field value are 'basket=BSK123', 'wave=20181011-123' and 'tmc=TMC_HSI/123'. In

[9] The pipe character is defined as "ASCII decimal 124, hex 7C"

[10] This field is not a direct equivalent to the order capacity tag in FIX protocol, In-Scope Brokers might be required to derive the orderCapacity as defined by the DS-OL using multiple internal values in their trading system

[11] For multi-leg orders or strategies, only those order legs of In-Scope Products are needed in the DS-OL format.

[12] The equal sign is defined as "ASCII decimal 61, hex 3D"

case an order is part of multiple collections (i.e. an order may appear in multiple groupings), the In-Scope Broker should use '|' as delimiter to concatenate them, e.g., 'basket=BSK123|tmc=TMC_HSI/123'.

3.1.1.5 Aggregated Orders

34. Aggregated orders (i.e., orders created in the In-Scope Broker's trading system by consolidating similar orders into a single order to be handled collectively) should also be recorded using the Order New event with the list of constituent orders and their respective quantities captured using the aggregatedOrders field.[13] Below is an example of how the aggregatedOrders field could be populated:

Table 3.1: Aggregated orders example in the DS-OL format

clientID	***logicalOrderID***	***orderQty***	***aggregratedOrders***
CLIENT-A	A01	1000	
CLIENT-B	B01	2000	
__na__	G01	3000	A01=1000|B01=2000

Note: Field clientID is not applicable (na) to an aggregated order.

3.1.1.6 Client ID and Client Type

35. There are four client types defined in the DS-OL. When an In-Scope Broker provides agency trading services to a securities broker that is a group company, this entity is an 'internal (institutional)' client ('INSI'), otherwise the entity is an 'external (institutional)' client ('INSE'). A 'principal' client ('PROP') will typically be a business unit within an In-Scope Broker or a group company, such as a client facilitation or proprietary trading desk. Client type ('RETL') is used for a retail client.

36. In general, the clientID is a unique identifier used by the In-Scope Broker to refer to the immediate client from which the order is received. Where the immediate client might be trading on behalf of another client, i.e. the end client, this 'end client' information is only required in some specific scenarios such as when an order is handled by multiple group companies and where the In-Scope Broker has knowledge of the end client's identity.

37. Where an In-Scope Broker has different internal proprietary trading desks, names of these desks can be used as clientID, and the Reference Data Dictionary can be used to provide additional details of these internal desks.

3.1.1.7 Event Response Type

38. In the DS-OL, both request and response part of an event are stored in the same record, and even if a trading system provides no message level response to a request, the eventResponseType field should still be populated

[13] In case the number of orders being aggregated is too large to be contained in the aggregatedOrders field, a 'Supplementary Event Record' can be used. Please refer to Section 3.4.2 for details.

with a logical response. I.e. As an order is accepted or rejected by an In-Scope Broker's trading system, either an 'ACK' or 'REJ' value should be set in the eventResponseType field.

39. In case an event does not have any logical response, e.g. if an order was sent to an Execution Venue but a response was never received (e.g., when the Execution Venue went down), this can be recorded with a 'No Response' ('__null__') value in the eventResponseType field.

40. In case of an Order Modify (please refer to Section 3.1.2 for details) or Order Cancel (please refer to Section 3.1.3 for details) event, if the modification or cancellation is initiated by the In-Scope Broker on behalf of the client, this is recorded with an 'Unsolicited' ('UNS') value in the eventResponseType field. Similarly, in a Split Modify (please refer to Section 3.1.5 for details) or Split Cancel (please refer to Section 3.1.6 for details) event, if the modification or cancellation of a split is initiated by an Execution Venue, the same principle applies.

3.1.2 Order Modify (OMOD) Event

41. The Order Modify event should generally be used when any Material Term of an order is modified. When multiple Material Terms of an order are modified in a single operation, only one Order Modify event is required. However, if multiple modify operations are performed separately on the same event, these Order Modify events should be recorded separately. For example, if a client order was originally traded as 'agency', but later the client requested for a guaranteed price of the whole order, the intended trading capacity should be modified to 'principal' using an Order Modify event.

42. As Execution Venue is not one of the Material Terms, no Order Modify event is required if an order is rerouted. E.g., if a client order was originally sent to an exchange, but later cancelled by an In-Scope Broker's trader and crossed in an ALP, such changes should be recorded as Split events (please refer to Section 3.1.4 for details).

3.1.3 Order Cancel (OCXL) Event

43. The Order Cancel events should be used whenever an order is cancelled. For mass cancel requests to exchanges in exception scenarios (e.g. a Kill Switch or Cancel on Disconnect), although these requests are not on order level, the result is that individual exchange-side orders are cancelled back to the In-Scope Broker's trading systems. Thus, these events should be recorded using Order Cancel events on the client orders as unsolicited cancels. A 'Mass Cancelled' flag (massCancelled) should also be used if the In-Scope Broker can provide this indicator.

3.1.4 Split New (SNEW) Event

44. The Split New event denotes a 'child order split' of a parent order. Throughout the order life cycle, regardless of the number of times the order is split while traversing through trading systems or technology stacks, only the order splits that may receive executions from Execution Venues are required to be

recorded with the Split New event. For the avoidance of doubt, child order splits that are eventually not executed are still required to be recorded. Child order splits for Internalised Trades are not strictly required if the orderCapacity of the child order split is the same as the parent order's orderCapacity.

45. For order handling which does not typically split an order into smaller orders and/or change any Material Terms before sending the order to an exchange for execution, a Split New event is not required.

3.1.5 Split Modify (SMOD) Event

46. The Split Modify event should be used when any Material Term of a child order split is modified. An example of a Split Modify event would be when an Algo child order split sent to an exchange has its order price modified by the Algo due to market movements.

3.1.6 Split Cancel (SCXL) Event

47. The Split Cancel event should be used when a corresponding child order split is cancelled, including unsolicited cancellations by an exchange. Please refer to Section 3.1.3 in respect of mass cancel requests sent to exchanges.

3.1.7 Execution (EXEC) Event

48. The Execution event denotes a 'trade report' for each client order from an Execution Venue. Execution events should be linked to corresponding Order events via LOID. Linkage to the immediate order or split order IDs is not required. If trade details are subsequently amended or cancelled, this should be recorded using an Execution Correction event (please refer to Section 3.1.8 for details).

3.1.7.1 Internalised Trades

49. Internalised Trades refer to:

- Trades that are executed internally, such as by internal crosses, ALP matches and manual fills; or

- Post-execution allocations of aggregated orders (please refer to Section 3.1.9 for details).

For these executions, the internalisedTrade field is set as '*true*'.

3.1.7.2 Agency Cross Trades

50. When an order is executed by crossing orders between agency clients, either through an ALP or via manual internal crossing, the crossTrade field should be set as 'true' with executionCapacity set as 'XA - Cross as Agency'. The counterpartyID field should indicate the actual counterparty of the cross in the corresponding Execution events on both sides.

3.1.7.3 Facilitated Trades from In-Scope Broker's Own Inventory

51. For Internalised Trades filled by the In-Scope Broker's own inventory, this should be denoted by an Execution event with the internalisedTrade field set as 'true'. The crossTrade field should be set as 'true' with executionCapacity set as 'XP - Cross as Principal'. The tradeBookID field should indicate the internal book of the In-Scope Broker which has provided the inventory.

3.1.7.4 Post-execution Allocations of Aggregated Orders

52. Where orders have been aggregated and executed orders need to be allocated to the constituent client orders, the client-side executions should be denoted with internalisedTrade set as 'true'. Also, the sourceExecVenue should provide the Execution Venue on which the market side executions have taken place. E.g. If the executionVenue field on an aggregated order shows XHKG, the executionVenue fields of the constituent orders should show the same.

3.1.7.5 Orders executed with mixed Execution Capacity

53. In the case of an Internalised Trade where the In-Scope Broker has assumed different types of execution capacity, the individual executions must be recorded separately. E.g., where a client order is partially filled by internal cross with an order from another agency client, the executionCapacity field should show XA; where this order is partially filled from the In-Scope Broker's own inventory, the executionCapacity field should show XP.

3.1.7.6 Odd Lot or Special Lot Executions

54. Odd lot or special lot fills are commonly provided to clients by In-Scope Brokers as Internalised Trades (internalisedTrade=true). In such cases, these trades should be recorded as principal cross trades (i.e., crossTrade=true, executionCapacity=XP), and with executionVenue=XOFF[14], oddLotTrade=true. The orderCapacity field should however remain as 'A' indicating that the intention of the order is to be traded as agency regardless of how that order is filled.

55. In case an odd lot or special lot is filled via an exchange facility directly (e.g. semi-automatic matching for odd lot or special lot via HKEX), this should be recorded as an exchange execution, i.e. crossTrade=false, executionCapacity=A, executionVenue=XHKG, and with oddLotTrade=true.

3.1.7.7 Execution Fields Examples

56. Below are some common scenarios and relevant fields:

[14] assume *'XOFF'* is used to denote an off-exchange execution venue

Table 3.2: Internalised Trade scenarios

Scenario	internalised Trade	cross Trade	executionCapacity	executionVenue	sourceExecVenue
A1	False	false	A – agency	<Exchange/Broker>	__na__
A2	False	false	P – principal	<Exchange/Broker>	__na__
A3	False	false	A – agency	<Exchange/Broker>	__na__
A4	False	true	XA – cross as agency	<ALP/Off-Exchange>	__na__
B1	True	true	XP – cross as principal	<Off-Exchange>	__na__
B2	True	true	XP – cross as principal	<Off-Exchange>	__na__
B3	True	false	A – agency	__na__	<Exchange/Broker>

- Scenario A1 - Agency orders execution at exchange/broker
- Scenario A2 - Principal orders execution at exchange/broker
- Scenario A3 - Agency orders execution in an aggregated order
- Scenario A4 - Cross trades / ALP matches (between agency clients)
- Scenario B1 - Facilitation trades (crossing between agency and principal)
- Scenario B2 - Odd lot portion fills (Internalised Trades)
- Scenario B3 - Post-execution allocations of aggregated agency orders

3.1.8 Execution Correction (EXCR) Event

57. The Execution Correction event denotes the 'corrected' execution details (i.e., executed price and/or quantity) on a summary basis. This event is applicable where the execution is either amended or cancelled and there should be only one single Execution Correction event per parent order which might consist of multiple child order splits and executions.

58. When performing execution amendment or cancellation to an order, In-Scope Brokers should not present the workflow specifics illustrating how executions are corrected but only the outcome of a correction. i.e., regardless of whether a correction is performed by (i) directly amending one or more existing execution details, (ii) allocating the executions first to an In-Scope Broker's error account and providing a new execution for the order, or (iii) via any other method. If a correction is performed more than one time, only the final correction of the day needs to be presented in the DS-OL format. There should be no additional Execution event for an order after the Execution Correction event.

59. By way of illustration, assume the following order executions and correction scenarios:

Table 3.3: Order executions

event	executionID	executionPrice	executionQty
EXEC	A11	1.1	1000
EXEC	A12	1.2	1000
EXEC	A13	1.3	2000
EXEC	A14	1.4	2000

Correction scenarios which are independent of each other:

- Scenario C1 - one summary of a new averaged price, no change in executed quantity
- Scenario C2 - cancelled all order executions for the client
- Scenario C3 - some executions prices are amended and some are cancelled,

Then the respective fields will be filled as in table 3.4:

Table 3.4: Different execution correction scenarios

Scenario	Event	executionID	executionPrice	executionQty	execCxlQty
C1	EXCR	AC11	1.15	6000	0
C2	EXCR	AC12	0	0	6000
C3	EXCR	AC13	1.16	2500	3500

3.1.9 Allocation (AALC) Event

60. The Allocation event should be used for ’aggregated orders’ only. When an aggregated order is partially or fully filled, these fills shall be allocated to the constituent client orders. Allocation events are recorded after all the Execution events of an aggregated order, with direct linkage to the client orders’ LOID using allocToLOID, and showing allocated quantity in the allocQty field. One Allocation event refers to one client order’s allocation, hence multiple Allocation events are expected for each aggregated order.

3.1.10 Order Summary (OSUM)

61. The Order Summary event is not part of the configuration of an order life cycle. Instead it is created to provide a record of the end of day summary of an order primarily for ease of identification and validation. Unlike the other events, Order Summary events are not necessarily the events which occur within a front-office system.

62. For each parent order completed within the same day, only one Order Summary event is required to provide information related to that order at the end of the order life cycle. Total executed quantity (totalExecutedQty) and averaged price (avgExecutedPrice) should be based on the final quantity and price to be settled with the client. For multi-day orders, however, an Order Summary is required which gives a summary of the multi-day order as of each corresponding end-of-day during its life cycle.

63. The orderCapacity field in an Order Summary event allows additional value 'AP' which can be used to indicate when a client order is intended to trade partially in an agency capacity and partially in a principal capacity.

64. Order Summary events are not required on aggregated orders but are required on the constituent client orders.

3.2 Reconstruction of Order Life Cycle

65. Order Life Cycle events, with the exception of Order Summary[15], can be assembled in different configurations to reconstruct many types of order life cycles or workflows.

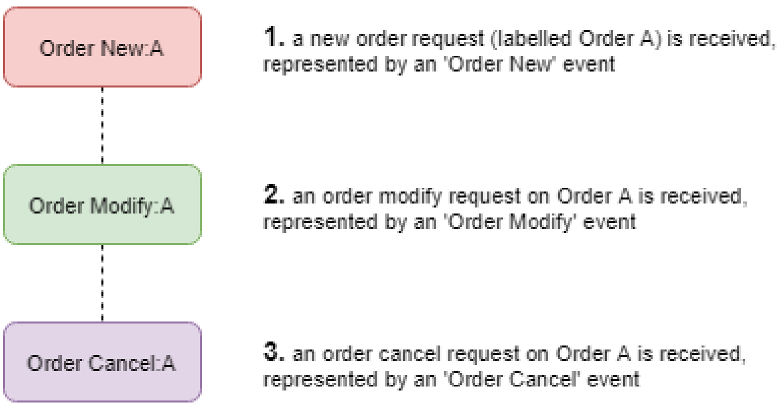
66. Order Life Cycles should be reconstructed using the following general guidelines:

- the 'first' or 'root' event is always an Order New (ONEW) event representing the order request as received (please refer to Section 3.1.1 for details), a unique *LOID* is assigned and is used as linkage information for all subsequent related events.

- if the order request is modified (including unsolicited modifies), an Order Modify (OMOD) event is used to denote the modify request (please refer to Section 3.1.2 for details).

- if the order request is cancelled (including unsolicited cancels), an Order Cancel (OCXL) event is used to denote the cancel request (please refer to Section 3.1.3 for details).

Figure 3.1: depicting the three types of Order Events, dotted line represents linkage of events with the same LOID

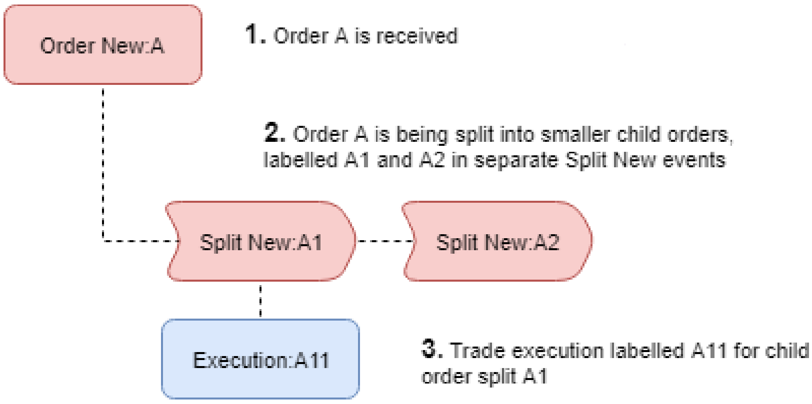


- Where an order is split into child orders and/or grandchild orders in a single system, only the child or grandchild order splits that are sent to

[15] For the purposes of illustration, the Order Summary event is not depicted in some of the examples.

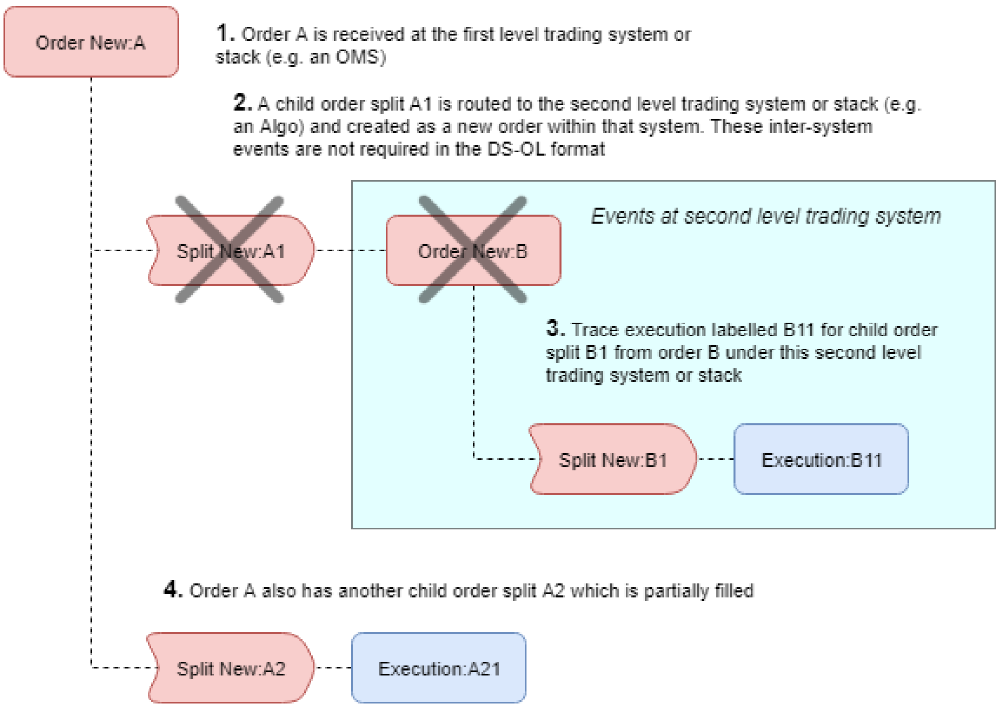
Execution Venues will be required to be recorded with Split New (SNEW) events.

Figure 3.2: depicting the child order splits of a parent order in a single system



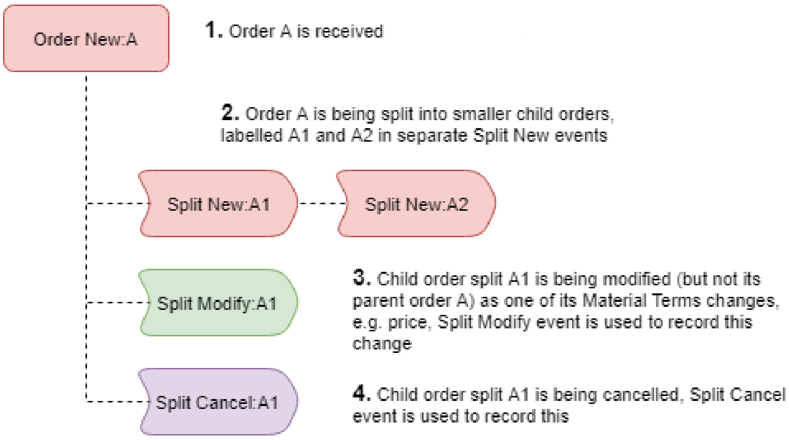
- When a client order is processed by more than one system, e.g., an order is received at a High Touch desk (first level system) before routing to an Algo for execution (second level system), the inter-system events are not required to be recorded in the DS-OL format.

Figure 3.3: depicting the child/grandchild order splits of a parent order in multiple systems



- Split Modify (SMOD) and Split Cancel (SCXL) events are used when the corresponding Split New events of child and/or grandchild orders that are required to be presented in the DS-OL format are being modified or cancelled respectively, including unsolicited modifications and cancellations (please refer to Sections 3.1.5 and 3.1.6 for details).

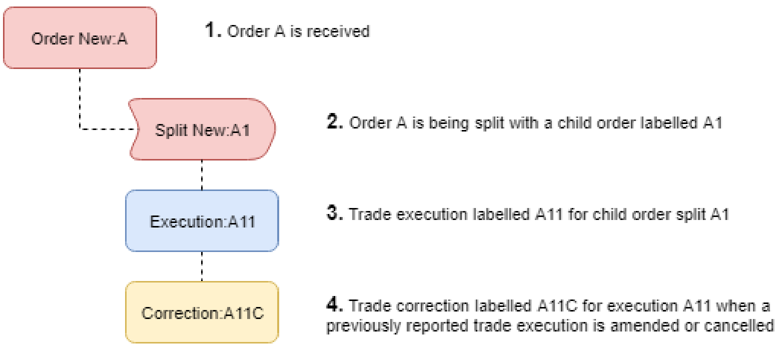
Figure 3.4: depicting the child order splits being modified and/or cancelled



- Execution (EXEC) events are used to record all trade executions from Execution Venues (please refer to Section 3.1.7 for details).

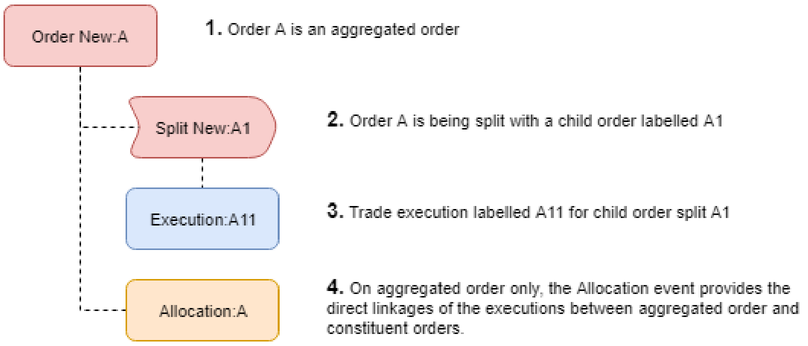
- For corrections on Execution events, Execution Correction (EXCR) events are used to record the new execution details (please refer to Section 3.1.8 for details). There should be no further Execution (EXEC) event after Execution Correction (EXCR) events.

Figure 3.5: depicting execution corrections



- Allocation (AALC) event, which is only used on aggregated orders, provides a direct linkage of the executions between the aggregated order and constituent client orders (please refer to Section 3.1.9 for details).

Figure 3.6: depicting post-execution allocation for aggregated orders



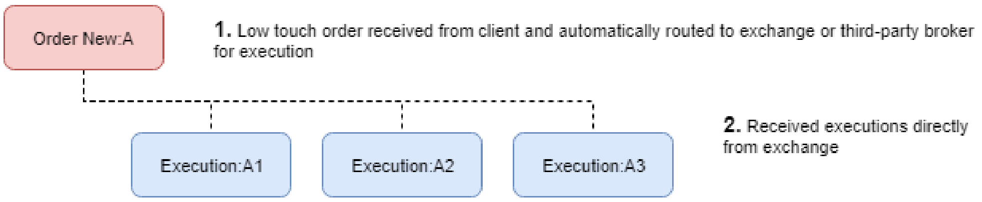
- an Order Summary (OSUM) event is used to provide a summary of an order and its execution price and quantity (except for aggregated orders, please refer to Section 3.1.10 for details).

67. Some common scenarios and example order flows are listed here for illustrative purposes.

3.2.1 Low Touch trading (without Algo)

68. For Low Touch trading[16] without Algo, only one Order New event is required when the Material Terms of an order are not modified when forwarded to an Execution Venue.

Figure 3.7: Low Touch trading orders without Algo routed directly to an Execution Venue with no change in Material Terms



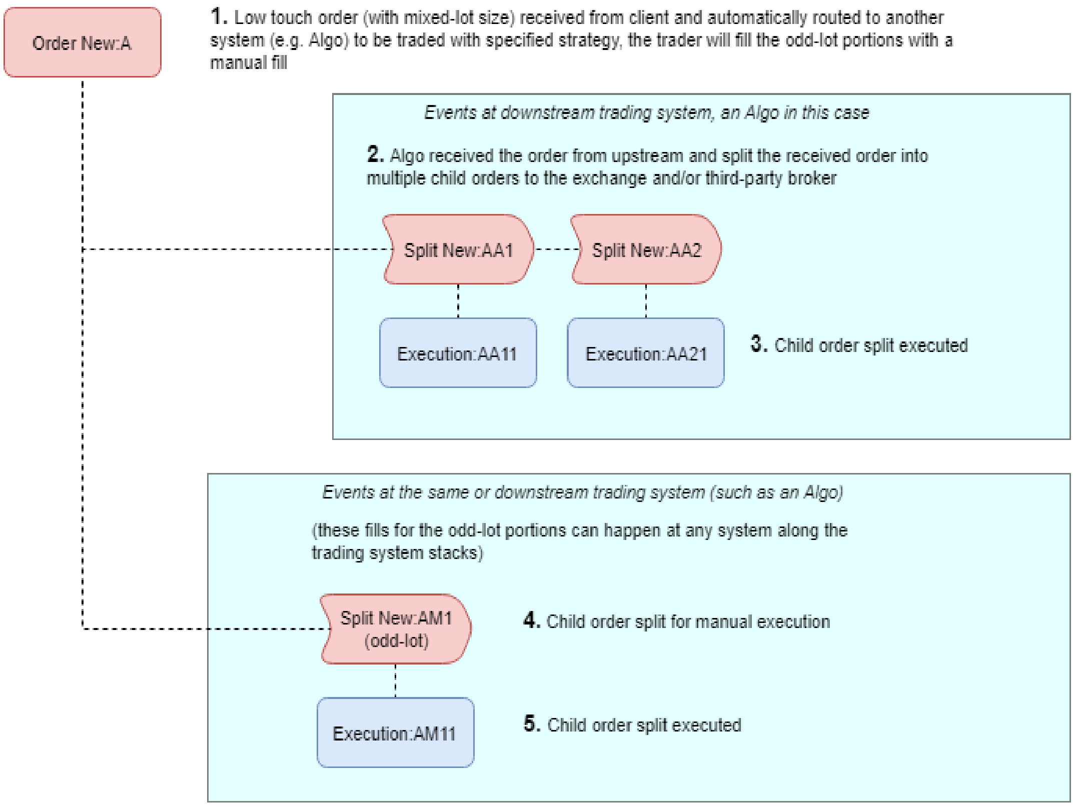
3.2.2 Low Touch trading (with Algo)

69. Similarly, for Low Touch trading with Algo, only one Order New event is required where orders are either routed initially through an OMS or directly to an Algo.

---

16 Low Touch trading refers to electronic trade flows that require little to no human intervention from order inception to processing, and eventually order amends, cancels and/or executions

Figure 3.8: Low Touch trading (with Algo)



3.3 Field Definitions

70. Throughout the DS-OL, fields will be defined per event type[17] and they are listed below:

Table 3.5: Fields in the DS-OL

Field Name	Data Type	Definition
event	eventType	The indicator of the event type as defined by the enumeration specifications
eventDateTime	timestamp	The creation time of the event. In-Scope Brokers should record the most accurate time they can get closest to the actual time
eventSequence	integer	This field is used in conjunction with the eventDateTime, if the timestamp in eventDateTime is not granular enough to accurately reconstruct the sequence of events.
eventRespDateTime	timestamp	The date and time that the acknowledgement or response was sent to the upstream system or to the client

[17] Some fields that are commonly used by In-Scope Brokers are also included in this table although they are only relevant to trading in futures contracts, options and certain types of derivatives, and not equities listed on the SEHK

Field Name	Data Type	Definition
eventResponseType	enum	Indicates the type of logical response of the recorded Order or Split event. If the Order or Split event has no actual response, use the 'null' filler (i.e. __null__).
eventRespText	string-extra	Text associated with an event response to record any additional details (e.g. the reject reason in the case of a rejection)
logicalOrderID	string-extra	An identifier assigned to the top parent order which should be unique throughout its life cycle.
orderID	string-long	The ID of the order generated by a system, this should be unique during a trading day but can be reused over time. In Split events, it is the order ID of the child order split itself. In Execution events, it is the order ID of the corresponding order or child order.
origOrderID	string-long	Indicates the original order ID on Modify event if the order ID changes after the modification. This field is not necessary if such order ID does not change after modification.
receivedOrderID	string-long	Indicates the order ID received from a client. For orders not created electronically or if the client side does not provide the order ID, this field is not required.
origReceivedOrderID	string-long	Indicates the original order ID received from a client in an Order Modify event if that received order ID changes after the modification. It is not needed if such order ID does not change after modification.
securityID	string-short	Indicates the security to be traded by the order, e.g. '0001.HK'
securitySource	enum	Specifies the type of the ID used in the securityID field, e.g. RIC, SEDOL, ISIN, etc. as defined by the Data Standards enumeration specifications.
securityType	enum	Indicates the type of security, e.g. equity, future, option, warrants/CBBCs, or others
accountID	dictionary key	The account ID assigned to an order. This must map to the In-Scope Broker's Reference Data Dictionary. Not required if the account is not known, or it is an aggregated order.
clientID	dictionary key	Key string used to identify the client defined by the In-Scope Broker. This must map to the In-Scope Broker's Reference Data Dictionary provided by the broker which may include other client-related identification such as LEI, entity name, or other detailed information about the client or internal desk. Not required if it is an aggregated order.
clientType	enum	Indicates the grouping of clients amongst 'Institutional (internal or external)', 'Proprietary/Principal' or 'Retail'.
orderCapacity	enum	In Order and Split events, it indicates the intended trading capacity of an order, i.e 'A -

Field Name	Data Type	Definition
		agency' or 'P - principal'. In Order Summary event, it also allows additional value 'AP - agency/principal' to indicate cases when an order is intended to trade partially in an agency capacity and partially in a principal capacity.
orderChannel	string-short	A free text field which indicates how an order is received, e.g. voice, electronic, internet, mobile, etc.
orderPrice	price	The price value for limit based order types, for market order types, a 'zero' value should be used
orderQty	number	For day orders, it is the quantity of the order. For multi-day orders, this can be the rolled over quantity from the previous trading day or the original order quantity. In Modify events, this indicates the intended or target order quantity of the modification, regardless of whether the order was partially filled or not.
orderType	enum	Indicates the type of order as defined by the enumeration specifications.
timeInForce	enum	Indicates the lifetime of the order, e.g. today only, valid until specified date, valid until it is cancelled or completed, as defined by the Data Standards enumeration specifications.
side	enum	Indicates whether the order is a Buy, Sell, Short Sell or Short Sell Exempt, etc. as defined by the Data Standards enumeration specifications.
shortLocateReq	boolean	For short sell orders only, this indicates whether the client has already covered the short amount, or needs the In-Scope Broker to locate the stock for borrowing.
currency	currency	The currency of the order, if it has been provided by the client
aggregatedOrders	key/value pairs	If multiple orders are aggregated into a new single order for trading, then this field contains the list of those orders' logicalOrderIDs and respective quantities which make up the aggregated order.
collectionID	key/value pairs	A free text field as a unique identifier for a 'collection of orders'. (Please refer to Section 3.1.7.4 for details.)
algoStrategyID	dictionary key	The key value of the referenced Algo strategy, e.g. 'VWAP', 'percentage of volume', etc. as defined by the In-Scope Broker.
algoAttributes	string-extra	Algo parameters as part of order requests as specified by the client. The In-Scope Broker can concatenate all specified parameters into a single string with proper delimiters.
altLiquidityInd	enum	Indicates whether a client order would participate in ALPs. In case not indicated in the client order or not defined in the client

Field Name	Data Type	Definition
		setup, this should default to the opt-in/opt-out value according to the client agreement with the In-Scope Broker.
automatedSplit	boolean	Indicates whether this child order split is generated automatically by a system. E.g., From an Algo the child order splits are normally generated automatically, hence this value will be set as 'true' for these child order splits. In another case, e.g. if a trader needs to fill the child order manually, this is not an automatic split and this value will be set as 'false'.
businessFlow	string-short	This free text field indicates the type of business or services required to be provided for effecting an order, e.g. High Touch, Low Touch, Portfolio, etc. This can be freely defined by the In-Scope Broker which best describes its business activities.
crossForbidden	boolean	'true' indicates that a client has requested that the order must be executed at an exchange.
directedOrder	boolean	Indicates whether the client has specified an Execution Venue for execution, e.g. for Low Touch orders, this would normally be 'true', but for High Touch this would normally be 'false'.
executionVenue	dictionary key	Key string for an internal mnemonic used by the In-Scope Broker for an Execution Venue, which can be industry standards such as ISO MIC code, or non-standard venue name as defined by the In-Scope Broker. In both cases, it should map to the In-Scope Broker's Reference Data Dictionary. This field is not applicable to allocation fills from aggregated orders.
expireDateTime	timestamp	Used in conjunction with GTD orders to indicate the order expiry date
orderInst	dictionary key	Indicates any In-Scope Broker defined order instruction that is not already captured in the orderType and/or algoAttributes fields. This must map to the In-Scope Broker's Reference Data Dictionary.
freeText	string-extra	Captures any free text fields on an order new or modify message. This usually contains information from client order instructions, or inter-desk instructions other than the data supplied in the orderInst field. For a child order split that does not have its own instructions, copy the value from its parent order if available.
systemID	string-short	A free text string name or identifier of the primary system recording the event, e.g. OMS, Algo, etc. as defined by the In-Scope Broker.
salesID	dictionary key	Indicates the name or identifier of the sales trader processing or monitoring the order.

Field Name	Data Type	Definition
		This must map to the In-Scope Broker's Reference Data Dictionary.
traderID	dictionary key	Indicates the name or identifier of the trader processing or monitoring the order. An In-Scope Broker can use a system user name (or a valid filler value) for automated trade processes such as low touch order. This must map to the In-Scope Broker's Reference Data Dictionary.
initiator	enum	Indicates the initiator of a modify or cancel event, e.g. 'Broker' is shown as the initiator when a partially-filled Algo order is cancelled by the Algo automatically because the market is closed whereas 'Client' is shown when a partially-filled Algo order is cancelled upon a client request.
massCancelled	boolean	'true' indicates that orders are cancelled due to mass cancel requests sent by the In-Scope Broker to an exchange in exception scenarios. This is not intended for client-side mass cancel request. Not required if this information is not captured by the In-Scope Broker.
nDayOrderQty	number	If In-Scope Brokers keep the same original orderQty for multi-day orders, this should be used to indicate the actual quantity for the rolled-over order at a given day.
solicitationType	enum	Indicates if an order, either agency or principal, is a result of IOI generated by the In-Scope Broker. If it is generated from a genuine client order, use 'AIOI', if it is generated by the In-Scope Broker's proprietary desk, use 'PIOI'.
internalisedTrade	boolean	'true' indicates an Internalised Trade.
tradeBookID	string-short	Indicates the code of the trading book that the position was booked under. For manually filled order, this field indicates the internal book used by the In-Scope Broker to manually fill the order from its own inventory.
counterpartyID	string-short	Indicates the counterparty for an internally crossed trade, which could be the internal clientID or an exchange 'broker code'.
crossTrade	boolean	'true' indicates that the trade is an internally crossed trade by the In-Scope Broker, either against another agency client or a proprietary desk for facilitated orders.
executionCapacity	enum	Specifies the capacity in which the execution was traded, e.g. 'agency', 'principal', 'cross as agency' or 'cross as principal'
executionID	string-long	An identifier generated by the In-Scope Broker for individual Execution or Execution Correction events. This field is not necessarily the execution ID provided by an exchange.

Field Name	Data Type	Definition
executionPrice	price	The price executed in the market, as a cross, or against an internal book. Use 'zero' in Execution Correction event when cancelling an execution for the client.
executionQty	number	In an Execution event, indicates the executed quantity at an Execution Venue. When used in an Execution Correction event, indicates the sum of executed quantity after the correction.
sourceExecVenue	dictionary key	Only required in client-side order executions for executions allocated from an aggregated order to constituent orders. Key string for an internal mnemonic used by the In-Scope Broker to refer to Execution Venue where the order is executed. E.g. if the Execution Venue of an aggregated order is showing executionVenue=XHKG, then the constituent orders should show sourceExecVenue=XHKG.
execCxlQty	number	Indicates the sum of executed quantity cancelled from a client order in an Execution Correction event
oddLotTrade	boolean	'true' indicates the execution is for an odd lot or a special lot.
tradeSession	enum	Indicates during which corresponding exchange trading session that off-exchange executions are effected by the In-Scope Broker, e.g. manual fills, internal crossing or ALP executions.
allocID	string-long	For aggregated orders only, an identifier generated by the broker for individual Allocation events.
allocToLOID	string-extra	For aggregated orders only, this is the logicalOrderID of the client order being allocated to.
allocQty	number	For aggregated orders only, this is the quantity being allocated to a particular client order.
optExpireDate	date	Options only: indicates the expiration date of an option.
optPutCall	enum	Options only: indicates if this is a put or call option.
optStrikePrice	price	Options only: indicates the strike price of an option.
underlyingIDSource	enum	Specifies the type of the identifier used in the underlyingSecurityID field, e.g. RIC, SEDOL, ISIN, etc., for derivative security such as futures contracts and options only.
underlyingSecurityID	string-short	Indicates the underlying security of an order, e.g. '0001.HK'. For derivative security such as futures contracts and options only.
initialOrderPrice	price	Indicates the initial order price, prior to any order modification in an Order Summary Event. For multi-day orders, the In-Scope Broker only needs to provide the initial order price when the original order is received.

Field Name	Data Type	Definition
initialOrderQty	number	Indicates the initial order quantity prior to any order modification in an Order Summary Event. For multi-day orders, the In-Scope Broker only needs to provide the initial order quantity when the original order is received.
finalOrderPrice	price	Indicates the final order price, after all order modifications in an Order Summary Event. For multi-day orders, the In-Scope Broker only needs to provide the final order price upon the completion of order execution.
finalOrderQty	number	Indicates the final order quantity after all order modifications in an Order Summary Event. For multi-day orders, the In-Scope Broker only needs to provide the final order quantity upon the completion of order execution.
avgExecutedPrice	price	The average price of all the executions on the order, it should be the average execution price eventually used to settle with a client
totalExecutedQty	number	The total quantity executed on the order, it should be the total quantity eventually used to settle with a client

3.4 Field Requirement Status by Event Type

71. Each field will be notated with M or IA to represent whether the field is Mandatory or If-Applicable.

Table 3.6: Field Requirement

Value	Abbreviation	Description
Mandatory	M	Required for the given event. This field must always be included.
If-Applicable	IA	Required for the given event unless it is not applicable to the scenario.

72. The table below summarises the requirement status of each field for each possible event type.[18]

[18] The 'Supplementary Event Record' is not an actual event.

Table 3.7: Field Requirement Status by Event Type

Field Name	Column	Data Type	ONEW	OMOD	OCXL	SNEW	SMOD	SCXL	EXEC	EXCR	AALC	OSUM
event	1	eventType	M	M	M	M	M	M	M	M	M	M
eventDateTime	2	timestamp	M	M	M	M	M	M	M	M	M	M
eventSequence	3	integer	IA	IA	IA	IA	IA	IA	IA	IA	IA	IA
eventRespDateTime	4	timestamp	IA	IA	IA	IA	IA	IA				
eventResponseType	5	enum	M	M	M	M	M	M				
eventRespText	6	string-extra	IA	IA	IA	IA	IA	IA				
logicalOrderID	7	string-extra	M	M	M	M	M	M	M	M	M	M
orderID	8	string-long	M	M	M	M	M	M	M			M
origOrderID	9	string-long		IA			IA					
receivedOrderID	10	string-long	IA	IA	IA							
origReceivedOrderID	11	string-long		IA								
securityID	12	string-short	M	M	M	M	M	M	M	M	M	M
securitySource	13	enum	M	M	M	M	M	M	M	M	M	M
securityType	14	enum	M									M
accountID	15	dictionary key	IA	IA		IA	IA					
clientID	16	dictionary key	M	M	M	M	M	M				M
clientType	17	enum	M									M
orderCapacity	18	enum	M	IA		M	IA					M
orderChannel	19	string-short	IA									
orderPrice	20	price	M	IA		M	IA					
orderQty	21	number	M	IA		M	IA					
orderType	22	enum	M	IA		M	IA					
timeInForce	23	enum	M	IA		M	IA					
side	24	enum	M			M			M	M	M	M
shortLocateReq	25	boolean	IA									
currency	26	currency	IA									
aggregatedOrders	27	key/value pairs	IA	IA								
collectionID	28	key/value pairs	IA	IA								
algoStrategyID	29	dictionary key	IA	IA		IA	IA					
algoAttributes	30	string-extra	IA	IA		IA	IA					
altLiquidityInd	31	enum	IA									
automatedSplit	32	boolean				M						
businessFlow	33	string-short	M									

Field Name	Column	Data Type	ONEW	OMOD	OCXL	SNEW	SMOD	SCXL	EXEC	EXCR	AALC	OSUM
crossForbidden	34	boolean	IA									
directedOrder	35	boolean	IA									
executionVenue	36	dictionary key				M			M	M		
expireDateTime	37	timestamp	IA	IA		IA	IA					
orderInst	38	dictionary key	M	IA		M	IA					
freeText	39	string-extra	IA	IA	IA	IA	IA	IA				
systemID	40	string-short	M	M	M	M	M	M	M	M	M	M
salesID	41	dictionary key	IA									
traderID	42	dictionary key	M	IA	IA	M	IA	IA	IA	IA		
initiator	43	enum		M	M		M	M				
massCancelled	44	boolean			IA			IA				
nDayOrderQty	45	number	IA									
solicitationType	46	enum	IA									
internalisedTrade	47	boolean							M	M		
tradeBookID	48	string-short							IA	IA		
counterpartyID	49	string-short							IA	IA		
crossTrade	50	boolean							M	M		
executionCapacity	51	enum							M	M		
executionID	52	string-long							M	M		
executionPrice	53	price							M	M		
executionQty	54	number							M	M		
sourceExecVenue	55	dictionary key							IA	IA		
execCxlQty	56	number								M		
oddLotTrade	57	boolean							IA	IA		
tradeSession	58	enum							IA	IA		
allocID	59	string-long									M	
allocToLOID	60	string-extra									M	
allocQty	61	number									M	
optExpireDate	62	date	M									
optPutCall	63	enum	M									
optStrikePrice	64	price	M									
underlyingIDSource	65	enum	IA									IA
underlyingSecurityID	66	string-short	IA									IA
initialOrderPrice	67	price										M

Field Name	Column	Data Type	ONEW	OMOD	OCXL	SNEW	SMOD	SCXL	EXEC	EXCR	AALC	OSUM
initialOrderQty	68	number										M
finalOrderPrice	69	price										M
finalOrderQty	70	number										M
avgExecutedPrice	71	price										M
totalExecutedQty	72	number										M

3.4.1 Field Filler Values

73. For all Mandatory (M) and If-Applicable (IA) fields per Event type, if a field value is not applicable or not required in a specific scenario, a 'filler' value must be provided unless otherwise specified. Two possible choice of filler value that can be used:

Table 3.8: Choice of Filler Value

Choice	Filler Value	Note
1	__na__	means not applicable, string 'na' prefixed and suffixed by two underscore characters
2	__null__	means there is no value or an empty value, string 'null' prefixed and suffixed by two underscore characters

If a field is neither Mandatory nor If-Applicable for an event, the field should be left blank.

3.4.2 Supplementary (ONEWS / OMODS / OCXLS / SNEWS / SMODS / SCXLS) Event

74. The Supplementary Event is not one of the order life cycle events. The supplementary Event record is only used to hold additional data for specific fields which can contain multiple delimited values where the total length of the data to be recorded exceeds the specified field length limit. To construct a supplementary record, please note the following:

- the event type field is denoted by appending supplemented event type with 'S'. E.g., ONEWS for an ONEW supplement
- the supplementary event record should have the same *logicalOrderID* as the supplemented event
- the supplementary event record should have the same timestamp as the supplemented event in *eventDateTime* field and use *eventSequence* field to denote the sequence
- individual value or key/value pairs must be kept as whole records in an event field (i.e., break the data only by removing a delimiter) Assuming a field can only hold up to 2 key/value pairs in the following example, the string 'key1=value1|key2=value2|key3=value3', should be broken up into 'key1=value1|key2=value2' and 'key3=value3'

- the supplementary event must be recorded on a line that immediately follows the line that records the supplemented event in the same data file provided to the SFC.
- there can be as many supplementary events as needed to hold all the data

Table 3.9: Supplementary Event Record

Field Name	Data Type	Description	Req.
Event	eventType	**ONEWS / OMODS / OCXLS / SNEWS / SMODS / SCXLS**	M
eventDateTime	timestamp	The creation time of the event. In-Scope Brokers should record the most accurate time they can get closest to the actual time, as long as all timed events can be properly ordered based on this timestamp.	M
eventSequence	integer	This field is used in conjunction with the eventDateTime, in order to accurately reconstruct the sequence of events if the timestamp in eventDateTime is not granular enough.	M
logicalOrderID	string-extra	An identifier generated by In-Scope Brokers which should be unique over an extended period of time and across trading days. This field links together all of the events in the order's life cycle, allowing easy analysis of the full hierarchy of order, child or grandchild order splits, executions, execution corrections and order summary events.	M
orderID	string-long	The ID of the order generated by a system, this should be unique during a trading day. Unlike logicalOrderID, it is not strictly required to be unique across trading days. In Split events, it is the order ID of the child order split itself. In Execution events, it is the order ID of the attached order or child order.	M
aggregatedOrders	key/value pairs	If multiple orders are aggregated into a new single order for trading, then this field contains the list of those orders' logicalOrderIDs and respective quantities which makes up the aggregated order.	IA
algoAttributes	string-extra	Algo parameters as part of Order Instructions as specified by the client. In-Scope Broker can concatenate all specified parameters into a single string with their choice of delimiters.	IA
freeText	string-extra	Captures any free text fields on an order new or modify message. This usually contains client order instructions, or inter-desk instructions. For child order splits that do not have its own instructions, copy the value from its parent order whenever possible.	IA

4. Technical Specifications: Data Format Specifications

4.1 Data Types

75. All submitted data will be validated based on the defined data type of each item, including formatting and range checking.

Table 4.1: Data Types in the DS-OL

Data Type	Description	Example
boolean	Either 'true' or 'false' value, lower cases without quotation marks	true
dictionary key	A dictionary maps the value as key to the corresponding metadata which describe the meaning of the value	refer to Section Reference Data Dictionary
enum	A value which is selected from a pre-defined list of options, as provided by the Data Standards	refer to Section 4.3
key/value pairs	A list of zero or more tuples where each tuple is comprised of a string key, a separator ('=', ASCII decimal 61, hex 3D), and a string value. Lists of more than one key/value pairs use the pipe character ('|', ASCII decimal 124, hex 7C) to separate tuples.	refer to Section Maximum Field Length
number	A numeric value	132.4412
integer	A numeric value with no decimal or fractional component	120000
price	A numeric value representing a price, maximum value allowed with 13 digits to the left of the decimal point and 10 digits to the right i.e. 9999999999999.9999999999	3.14159
string-short	Free text field with a maximum length of 64 chars For a list of string values use the pipe character ('|', ASCII decimal 124, hex 7C) to separate individual strings.	refer to Section Maximum Field Length
string-long	Free text field with a maximum length of 256 chars For a list of string values use the pipe character ('|', ASCII decimal 124, hex 7C) to separate individual strings.	refer to Section Maximum Field Length
string-extra	Free text field with a maximum length of 2048 chars For a list of string values use the pipe character ('|', ASCII decimal 124, hex 7C) to separate individual strings.	refer to Section Maximum Field Length
timestamp	Timestamps are represented in UTC YYYYMMDD-HH:mm:ss.nnnnnnn Valid values:	20180521-09:00:00 20180521-

Data Type	Description	Example
	- YYYY (Year) = 0000-9999 - MM (Month) = 01-12 - DD (Day) = 01-31 - HH (Hour) = 00-23 (24-hour notation) - mm (Minute) = 00-59 - ss (Second) = 00-59/60 (60 only if UTC leap second) Granular timestamp accuracy supported: - nnn (millisecond accuracy) - nnnnnn (microsecond accuracy) - nnnnnnn (100 nanosecond accuracy)	09:00:00.123
date	A calendar date format as defined by ISO-8601. i.e. YYYYMMDD	20180521
currency	3 letter currency code as defined by ISO-4217. Including non-ISO currency code 'CNH' and 'RMB'	HKD

4.2 Maximum Field Length

76. For each field stored in the file, there is a maximum length for each field depending on the data type. In-Scope Brokers should include validations to the data submitted to avoid sending data over the maximum length.

Table 4.2: Maximum Field Length

Data Type	Maximum length in characters
dictionary key	32
string-short	64
string-long	256
string-extra	2048
key/value pairs	32767
number, integer, price	24 (including decimal point, if any)
boolean	5
timestamp	25
date	8
currency	3

4.3 Enumerations

77. The following enumerations are used in corresponding order life cycle event data fields:

Table 4.3: Enumerations

Field Name(s)	Short Description	Valid Values	FIX Tag reference (if any)
altLiquidityInd	Intention to participate in ALP	0 = Opted-out 1 = Opted-in	
clientType	Type of the client	INSI = Institutional (internal or clients that are group companies) INSE = Institutional (external) PROP = Proprietary or principal RETL = Retail	
eventResponseType	Type of response to an event to the originator	ACK = Acknowledged / Accepted REJ = Rejected UNS = Unsolicited (use the '__null__' filler if the event did have no response)	Tag150 (extended to include Unsolicited)
eventType	Designated event type	ONEW = Order New OMOD = Order Modify OCXL = Order Cancel SNEW = Split New SMOD = Split Modify SCXL = Split Cancel EXEC = Execution EXCR = Execution Correction AALC = Allocation (for aggregated orders) OSUM = Order Summary - Supplementary event record types: ONEWS = Order New - Supplementary OMODS = Order Modify - Supplementary OCXLS = Order Cancel - Supplementary SNEWS = Split New - Supplementary SMODS = Split Modify - Supplementary SCXLS = Split Cancel - Supplementary	
executionCapacity	Traded capacity of an execution	A = Agency P = Principal XA = Cross as Agency XP = Cross as Principal	Tag29 (values are remapped)
initiator	Indicate who initiated an event	B = Initiated by others except client C = Initiated by client	
optPutCall	Put or Call option	0 = Put 1 = Call	Tag201

Field Name(s)	Short Description	Valid Values	FIX Tag reference (if any)
orderCapacity	Intended trading capacity of an order	A = Agency P = Principal AP = partially in Agency and partially in Principal (used only in Order Summary event)	
orderType	Type of an order	MKT = Market Order LMT = Limit Order	
side	Side of an order	1 = Buy 2 = Sell 5 = Sell short 6 = Sell short exempt	Tag54
solicitationType	Type of solicitation	AIOI = Natural IOI from client or facilitation (agency) PIOI = Not natural IOI from In-Scope Broker (principal)	Tag130 (simplified)
securityType	Type of a security	S = Stocks F = Futures O = Options W = Warrants or CBBCs Z = Others	Tag167, Tag461 (simplified)
timeInForce	Time in Force	0 = Day (or session) 1 = Good Till Cancel (GTC) 2 = At the Opening (OPG) 3 = Immediate or Cancel (IOC) 4 = Fill or Kill (FOK) 5 = Good Till Crossing (GTX) 6 = Good Till Date 7 = At the Close 9 = At Crossing	Tag59
tradeSession	Trading session of the execution	PO = Pre-opening session CT = Continuous trading session CA = Closing auction session OF = Off exchange hours	

4.4 Symbology

78. The DS-OL do not mandate the symbology, i.e. the use of ISIN, RIC, SEDOL, CUSIP, etc. to identify instruments, to be used by In-Scope Brokers. In-Scope Brokers are only required to be consistent in the symbology used within datasets provided to the SFC. e.g. if RIC is used as the symbology for equities, it should be used in all of the data recorded including the underlying instruments of futures contracts and options data.[19]

[19] For listed securities, it is recommended to use the symbology of the primary listing exchange.

Table 4.4: Symbols

Field Name(s)	**Short Description**	**Valid Values**	**FIX Tag reference (if any)**
securitySource	Source type of a security ID	1 = CUSIP 2 = SEDOL 3 = QUIK 4 = ISIN number 5 = RIC code 8 = Exchange Symbol A = Bloomberg Symbol UNK = Unknown Symbol (only for unresolved instruments)	Tag22 (extended to include unknown symbol)
underlyingIDSource	Source type of a underlying security ID	1 = CUSIP 2 = SEDOL 3 = QUIK 4 = ISIN number 5 = RIC code 8 = Exchange Symbol A = Bloomberg Symbol UNK = Unknown Symbol (only for unresolved instruments)	Tag305 (extended to include unknown symbol)

5. Reference Data Dictionary

79. In-Scope Brokers must provide their Reference Data Dictionary to the Commission to supplement order life cycle data. The Reference Data Dictionary is expected to contain the following details:

5.1 Account Details

Table 5.1: Account Details

Field Name	Data Type	Description	Example Value(s)
accountID	dictionary key	The key value of the referenced accountID in the corresponding Data Standards Events	01-2345678, AC123456
dateOpen	date	Account opened since date	20181231
dateModified	date	Account details modified date	20170101
accountType	enum	Type of the account in accountID	
accountDesc	string-extra	The detail description of the account nature and usage	Omnibus account for clients of ABC, plc. (U.K. affiliate) to trade Hong Kong stock
accountOwner	string-short	Name of the person(s) who own this account, if applicable	John Doe

5.2 Client Details

Table 5.2: Client Details

Field Name	Data Type	Description	Example Value(s)
clientID	dictionary key	The key value of the referenced clientID in the corresponding Data Standards Events	XYZAM
dateOpen	date	Client onboarded or started trading since date	20181231
dateModified	date	Client details modified date	20170101
entityName	string-long	Full name of the client or entity	XYZ Asset Management, Jane Doe, etc
entityDocType	enum	The client's or entity's identification document type, if applicable	Please refer to Enumerations tab
entityDocNumber	string-short	The client's or entity's identification document number, if applicable	X123456(9)
issuerCountryCode	string-short	Any ISO country code of document issuer, if applicable	HKG

5.3 User Details

80. These refer to personnel involved in the business operations under an In-Scope Broker; they are applicable to tradeID and salesID fields.

Table 5.3: User Details

Field Name	Data Type	Description	Example Value(s)
userID	dictionary key	The key value of the referenced traderID or salesID in the corresponding Data Standards Events	jdoe, algoagent1
dateOpen	date	User enabled since date	20181231
dateModified	date	User details modified date	20170101
userName	string-short	Short or Long name of the user, including actual and system users	John Doe, Algo Agent
userDept	string-short	Short or Long name of the department or desk in which the user belongs to, if applicable	Sales, Electronic Equity Trading
userRole	string-short	Short or Long name of the user's role, if applicable	Sales, Trading, etc

5.4 Execution Venue

81. In-Scope Brokers can use ISO standard MIC codes (e.g. XHKG for HKEX) or define their own mapping which are applicable to executionVenue and sourceExecVenue fields.

Table 5.4: Execution Venue

Field Name	Data Type	Description	Example Value(s)
executionVenue	dictionary key	The key value of the referenced executionVenue in the Data Standards Events, an internal mnemonic used by the In-Scope Broker for the venue	XHKG, SEHK, HKE, XOFF, BROKERX, POOL1, DARKPOOLX, etc.
executionVenueName	string-short	The full name or description of the execution venue	Hong Kong Stock Exchange, Off-exchange Manual Fill, Broker X, Dark Pool X, etc.

5.5 Algo Strategy

82. Where references are made to Algo, In-Scope Brokers should specify a reference document which outlines the strategy details.

Table 5.5: Algo Strategy

Field Name	Data Type	Description	Example Value(s)
algoStrategyID	dictionary key	The key value of the referenced Algo strategy, e.g. 'VWAP', 'percentage of volume', etc., as defined by the In-Scope Broker	VWAP
algoDocRef	string-long	The file name or URI of the document outlining the Algo details	VWAP.pdf, Algo-Specification.pdf

5.6 Order Instruction

83. In-Scope Brokers should describe the order instruction key for conventional instructions or instructions as defined by them. Each order instruction must specify its full name and/or detailed description of the instruction.

Table 5.6: Order Instruction

Field Name	Data Type	Description	Example Value(s)
orderInst	dictionary key	The key value of the order instruction provided in orderInst field	CD, enhanced-limit
orderInstDesc	string-long	The full name and/or detail description of the order instruction	CD order, HKEX enhanced limit order

5.7 Data Reference Enumerations

84. Only the following enumerations should be used in a Reference Data Dictionary:

Table 5.7: Data Reference Enumerations

Field Name(s)	Short Description	Valid Values	Reference (if any)
accountType	Type of the account	5 = Investor ID 24 = Customer Account	Tag1, Tag452
entityDocType	Type of the entity identifier or identity document	1 = Identity Card (e.g. HKID, ID issued by the relevant government authority) 2 = Passport 3 = Certificate of Incorporation or other official incorporation documents of the entity 4 = LEI 5 = Other official identity document of the individual	

Appendix – Data Reporting Instructions and Validation Guidelines

Data Submission format

1. The DS-OL datasets are to be provided as text-based files.

File Format

- To support double-byte characters (e.g. Chinese characters), the DS-OL data files are UTF-8 encoded. Data files *MUST NOT* add a byte order mark to the beginning of the DS-OL file
- Each line must contain one complete order life cycle event record
- Each record's fields are separated with a field delimiter (ASCII decimal 31, hex 1F[20])

File Naming

1.1. Order Life Cycle Data

Filenames are of the form:

<Date>_<CE>_DSOL_<File Number>.csv[.<compression extension>]

- **<Date>** is in the form: YYYYMMDD
- **<CE>** Registered Central Entity Number
- **<File Number>** is in the form: ZZZZZZ
  The file number is a 6-digit, left-padded with zero sequence number to allow splitting of daily data files across a number of files.
- **[.<compression extension>]** is an optional compression extension (.gz, .zip)

Table A1.1: Data File Name examples

Description	Example filenames
Single-file daily dataset	20180901_ABC123_DS0L_000001.csv
Multi-file daily dataset	20180901_ABC123_DS0L_000001.csv 20180901_ABC123_DS0L_000002.csv 20180901_ABC123_DS0L_000003.csv
Multi-day dataset	20180901_ABC123_DS0L_000001.csv 20180902_ABC123_DS0L_000001.csv 20180903_ABC123_DS0L_000001.csv
Single-file daily dataset compressed	20180901_ABC123_DS0L_000001.csv.zip

[20] As this field delimiter is a non-printable character, 'caret' symbol is used to represent this field delimiter for illustrative purposes in the Appendix.

1.2. Reference Data

Filenames are of the form:

<Date>_<CE>_DSOL_REF_<Ref Type>.csv[.<compression extension>]

- **<Ref Type>** is a choice of: **ACCT** (Account), **CLNT** (Client), **USER** (User), **EXCV** (Execution Venue), **ALGO** (Algo Strategy) or **ORDI** (Order Instruction)
- **<Date>** is in the form: YYYYMMDD
- **<CE>** Registered Central Entity Number
- **<File Number>** is in the form: ZZZZZZ
  The file number is a 6-digit, left-padded with zero sequence number to allow splitting of daily data files across a number of files.
- **[.<compression extension>]** is an optional compression extension (gz, zip)

Table A1.2: Reference Data File Name examples

Description	Example filenames
Account Details	20180901_ABC123_DS0L_REF_ACCT.csv
Client Details	20180901_ABC123_DS0L_REF_CLNT.csv
User Details	20180901_ABC123_DS0L_REF_USER.csv
Execution Venue	20180901_ABC123_DS0L_REF_EXCV.csv
Algo Strategy	20180901_ABC123_DS0L_REF_ALGO.csv
Order Instruction	20180901_ABC123_DS0L_REF_ORDI.csv
Client Details compressed	20180901_ABC123_DS0L_REF_CLNT.csv.zip

2. Data File Header

Files should start with a header line as the first line of the record in the following format:

Table A2.1: Data File Header

Field Name	**Data Type**	**Description**	**Example**
recordType	string-short	Use string 'HEAD' to denote this is a header line	HEAD
version	string-short	Version of the DS-OL adopted	DSOL-1.0
submitRef	string-short	Unique reference assigned by In-Scope Broker for the submission	20210910-00001
submitParty	string-short	Name of the In-Scope Broker submitting the data	XYZBANK
ceNumber	string-short	Registered Central Entity Number	XYZ123
startDate	date	Starting date of data submission included in this file	20210102
endDate	date	Ending date of data submission included in this file	20210331
totalNumRecords	number	Total number of event records in the submission file	83123

Example of the header line in the file:

```
HEAD^DSOL-1.0^20210910-00001^XYZBANK^XYZ123^20210102^20210331^83123
```

Please refer to Appendix Section 1 on field delimiter requirements.

3. Data File Contents

The contents of the file after the header line start with one line of delimited field names, then continues with one line per order life cycle event. The fields must be listed in the sequence as listed in table 3.16, if a field is not applicable (i.e. neither Mandatory nor If-Applicable) to the corresponding event, e.g. executionPrice is not applicable to an 'Order New' event, the field can be left blank.

Example of the delimited field names in the order life cycle data file, (only the first 10 fields are shown below for the sake of simplicity):

```
event^eventDateTime^eventSequence^eventRespDateTime^eventResponseTyp
e^eventRespText^logicalOrderID^orderID^origOrderID^receivedOrderID^.
...
```

Please refer to Appendix Section 1 on field delimiter requirements.

Example of the delimited field names in the reference data file:

```
accountID^dateOpen^dateModified^accountType
01-2345678^20181231^20170101^24
01-3344556^20121120^20140512^24
```

Please refer to Section 1 on field delimiter requirements.

Validation

4. The following sections are general guidelines that In-Scope Brokers can use to validate the contents of the DS-OL formatted data. Please note that examples given in these guidelines are not meant to be exhaustive.

Level 1 Validations

Table A4.1: Level 1 Validation

	Type of Check		Validation Guideline	Applicable Fields (for examples)
1.1	Data type validation	1.1.1	Verify the value of a field conforms to the required data type, e.g. with 'date' data type value should have 'date' component only but no 'time' component	All fields
1.2	Data range and constraints validation	1.2.1	Verify the value of a field is within range normally accepted by the content of the data, e.g. with 'price' or 'number' data type data should only have numeric values with a range limit	All fields
		1.2.2	Verify the value of an enumeration field only use valid values as defined by the data standards. Exception is allowed only if the value used is to represent concept not already pre-defined by the enumeration field	Enumeration fields
		1.2.3	Verify the value do not contain character to be used as data 'delimiter' when data may be exported to a file	All fields
		1.2.4	Verify proper separators are used when multiple values need to be stored in one field such as Key/Value Pairs fields and String fields that allows multiple values	Key/Value Pairs fields (e.g. *aggregatedOrders*)
1.3	Field requirement validation	1.3.1	Verify 'Mandatory' (M) fields in corresponding events do have data values, if value is empty or not applicable in specific case, a filler value is required	Mandatory fields
		1.3.2	Verify 'If-Applicable' (IA) fields in corresponding events do have data values if applicable, if not, a filler value is required	If-Applicable fields

Level 2 Validations

Table A4.2: Level 2 Validation

	Type of Check		Validation Guideline	Applicable Fields (for examples)
2.1	Cross reference validation	2.1.1	Linkage fields used to link orders between systems should be properly linked, e.g. logicalOrderID should be the same for all linked events, logicalOrderIDs in aggregatedOrders should be referring to corresponding logicalOrderIDs	event, logicalOrderID, aggregatedOrders, allocToLOID
2.2	Referential integrity check	2.2.1	For fields with 'dictionary key' data type, verify the values are those used as key to the linked reference data	accountID, clientID, executionVenue, sourceExecVenue, salesID, traderID, algoStrategyID
2.3	Record counts and statistics	2.3.1	Count the number of events by event types and check if the distribution is reasonable according to the business flows	event, businessFlow
		2.3.2	Compare the number of events between different event types and check if the ratio is reasonable, e.g. in total there should be more Split New events than Order New events	event
2.4	Consistency check	2.4.1	Check for data field consistency between events, e.g. the total executed quantity in the Order Summary should match with the corresponding Execution events and not larger than the order size (assume no overfills)	orderQty, nDayOrderQty, executionQty, execCxlQty, finalOrderQty, initialOrderQty, totalExecutedQty
		2.4.2	For Order/Split Modify events, if order ID do change after modification, check the original IDs are correct	origOrderID, origReceivedOrderID
		2.4.3	Events should be reconstructed according to the Technical Specifications, e.g. There is always only one ONEW event for a client order; There should be no additional EXEC event for an order after the EXCR event, etc.	event, logicalOrderID